

PRAGMA

A publication of Semaphore Corporation

Issue Number 5

August, 1983

IN THIS ISSUE

A Day of Revelation	14
Vanilla, Part 5: Receiving	20
An Introduction to ENGLISH®, Part 4: Being Choosy	26
Converting Manual Paint to Programs	28
A Program that Reports File Pointer Locations	32
Sysmap, Part 5: Automation	36
Permuted Index to the First Four Issues of Pragma	42

DEPARTMENTS

Utilities	6	Command Files	33
Producer Profile	10	Queries	35
Benchmarks	18	The Computer Room	36
Wish List	24	Letters	39
Local User Groups	30	Games	51

Your PC can now be a "Personal Minicomputer" with the **Revelation Operating System** from Cosmos.

Revelation is a Relational Data Base Management System for the Personal Computer that provides minicomputer power at microcomputer prices. You can now have a small, portable computer that will serve as a working tool on the job, at home or on the road for less than \$5,000.

Revelation is PICK™ compatible and operates on the IBM™ Personal Computer, as well as the Columbia Multi-Personal™ Computer and the COMPAQ™.

As a Relational Data Base Management system, **Revelation** operates on top of MS-DOS™, allowing you to switch back and forth between either operating environment and utilize hundreds of existing programs.

Relational Data Base Management and **Revelation** provide easy access to your data on floppy or hard disk. You use meaningful field names to refer to the data you desire. You can also define relationships that exist between two or more fields.

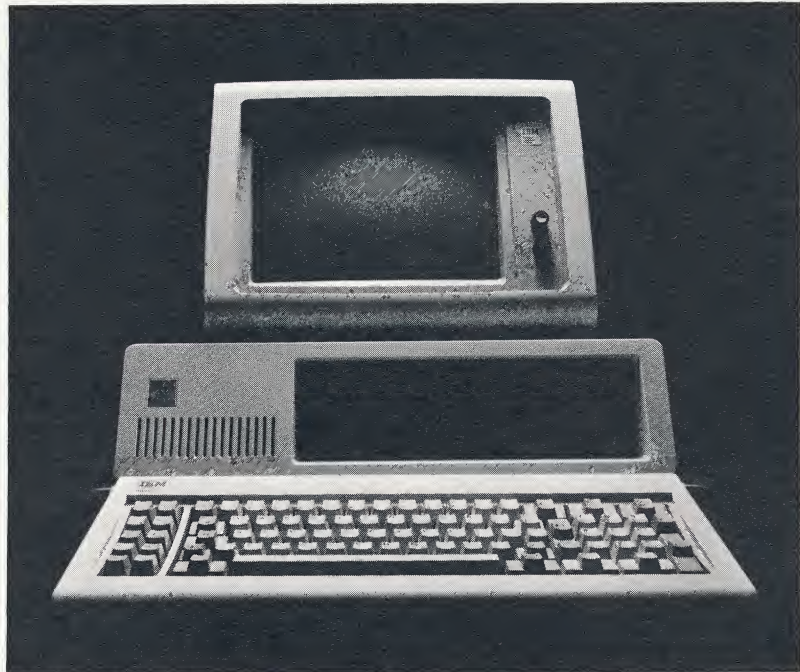
Revelation employs the Intel 8087 math chip on the PC for powerful number crunching capacity, floating point arithmetic and hardware mathematical functions.

Our application documentation and generation tools make **Revelation** ideal for software development and maintenance. And **Revelation's** advanced technology includes source code generation so you can do any job that needs to be done.

Communications problems between minicomputers and microcomputers are solved as **Revelation** allows your machine to communicate with compatible minicomputers.

Now you can afford the advantages of distributed processing.

Why wait?



IBM Personal Computer

Collect data at a remote site during the day and ship it to your minicomputer when you want. You can also use your PC as a dumb terminal to initiate jobs on your local or remote minicomputer.

System Memory hassles are eliminated. Just add one of the many available 512K RAM boards to your microcomputer and you have the memory you need to do your work quickly and efficiently with **Revelation**.

Now you can have minicomputer capabilities at microcomputer prices. From Cosmos.

Sample configuration

Sample IBM System

- System unit, 64K RAM keyboard, 160K disk drive, and disk drive adapter

System price

• 2nd 160KB disk drive	\$2,205
• Monochrome display	\$450
• Monochrome display and printer adapter	\$345
• 8087 math chip	\$335
• 256K RAM with serial interface	\$220
• Revelation software	\$349
• MS-DOS	\$950
Total	\$4,894

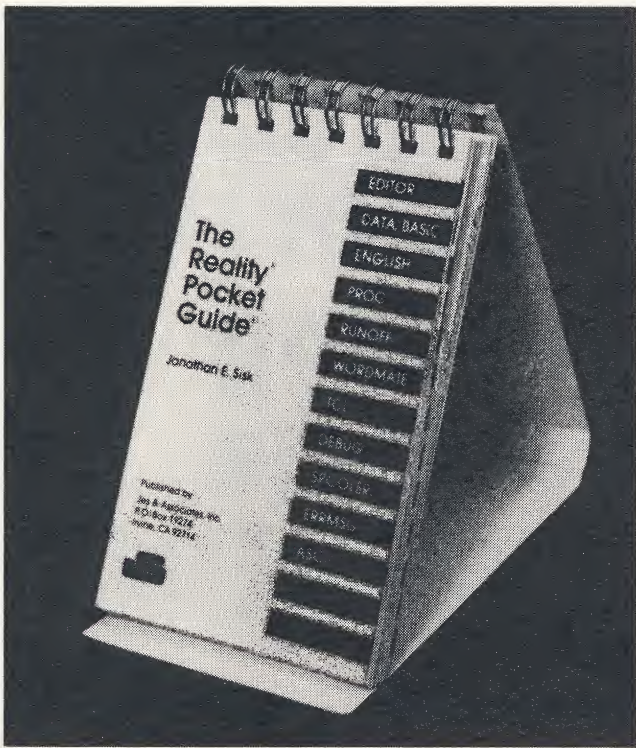
Minimum configuration

- 320K RAM
- 8087 math chip
- MS-DOS or PC-DOS
- One single-sided floppy drive

COSMOS

P.O. Box AH
123 Ferntree Drive W.
Morton, WA 98356
(206) 496-5974
24-hour answering service: (206) 226-9362

MS/DOS TM of Microsoft Corp.; Intel TM of Intel Corp.; ADDS Mentor TM of Applied Digital Data Systems; PICK Operating System TM of PICK SYSTEMS; COMPAQ Portable Computer TM of COMPAQ Computer Corp.; Columbia Multi-Personal Computer is registered TM of Columbia Data Products Inc.



THE NEW SOFT COVER EDITION

OUR PRODUCTS MAKE YOUR COMPUTER EASIER TO USE.

From Jonathan E. Sisk, author of the original "REALITY POCKET GUIDE" and the "PICK POCKET GUIDE" comes the next generation of quick-reference products that describe every command pertaining to the following topics:

- EDITOR
- DATA/BASIC
- ENGLISH
- PROC
- RUNOFF
- WORDMATE
- TCL
- SYSTEM DEBUGGER
- DATA/BASIC DEBUGGER
- SPOOLER
- ERRMSG
- ASCII

- **THE REALITY POCKET GUIDE, Hard-Cover Edition** **\$29.95**
The original, now in its third printing, is current with Release 4.2. The 6-ring binder fits easily into a briefcase and lays flat on any surface for easy use. Colored index tabs make access to any section a snap.
- **THE REALITY POCKET GUIDE, Soft-Cover Edition** **\$19.95**
(SHOWN ABOVE). The same contents as the hard-cover edition, but packaged in a self-standing spiral binder.
- **THE REALITY HELP PROCESSOR** **\$95.00**
The on-line version of the REALITY POCKET GUIDE provides instant access to over 600 help screens by simply entering the word, HELP, followed by the desired command. Each help screen displays the format of a command and a brief explanation of its use. Provided on a half-inch tape.
- **THE PICK HELP PROCESSOR** **\$95.00**
The on-line version of the PICK POCKET GUIDE, for users of any system running the PICK Operating System. Same functions as the REALITY HELP PROCESSOR. Provided on a half-inch tape.

Product	Unit Price	QTY.	Extended Price
Reality Pocket Guide, Hard-Cover Edition	\$29.95	x _____ = _____	
Reality Pocket Guide, Soft-Cover Edition	19.95	x _____ = _____	
Reality Pocket Guide, Edition 3 Update	10.00	x _____ = _____	
Reality Help Processor	95.00	x _____ = _____	
Pick Help Processor	95.00	x _____ = _____	

Shipping, Per Order

Books, Within United States	\$ 2.50		_____
Books, Outside United States	7.50		_____
Tapes, Within United States	10.00		_____
Tapes, Outside United States	25.00		_____

California State Tax (6%) _____

Total Enclosed _____

Company _____
Address _____
City _____ State _____ Zip _____
Phone _____ Contact _____
Computer Manufacturer _____ 800/1600 BPI _____

Please make checks payable to:



JES & Associates, Inc.
P.O. Box 19274
Irvine, CA 92714
(714) 786-2211

Sorry, COD's Not Accepted.
Allow 2 to 4 Weeks for Delivery.
Quantity Discounts Available. Information on Request.

PROC
PROC TRANSFERS

In conjunction with the special feature on menus, this column will discuss Proc transfers. The Proc transfer feature is one of the handiest commands available through Proc, and is useful for far more than just menu construction. They can be used for jobstreams and linking Procs together as well.

A Proc transfer is simply a way to branch to another Proc on any file in the system. There are two types of Proc transfers, temporary and permanent transfers. During the running of a Proc, if a Proc transfer is encountered, control is passed to the Proc specified in the transfer (one Proc per transfer). If the transfer is a permanent one, then control remains with this second Proc. If the transfer is a temporary one, then control is returned to the original Proc when the second Proc is finished.

A Proc transfer looks like this:
(PROCFILE CUSTOMER.REPORT)

This is a permanent transfer and will pass control to a Proc called 'Customer.Report' in a file called 'Procfile'.

(PROCFILE CUSTOMER.REPORT)

This is a temporary transfer. It's enclosed in brackets, while the permanent transfer is enclosed in parenthesis.

This temporary Proc transfer will pass control to the same 'Customer.Report' Proc but after the 'Customer.Report' is finished, it will return back to the original Proc.

In order to perform this 'return' to the original Proc, the 'Customer.Report' Proc must have a 'RPN' statement at the end of it. When the 'RPN' statement is encountered, control is returned back to the original Proc.

A 'RPN' statement is harmless and can be put in any Proc, regardless of whether or not it will be called by a Proc transfer. This allows Procs to be used by themselves or called by other Procs.

An example will help to illustrate the Proc transfer better. The first Proc ('Sales.Report') in the example below contains a Proc transfer on line 6. When the Proc hits this line, control is passed temporarily to the second Proc ('Customer.Report').

AN EXAMPLE
OF A
PROC
TRANSFER

SALES.REPORT

001 FO

002 H SORT SALES-FILE BY DATE

PAGE 19

The solution to those confusing reference manuals!

FROM INFORMATION RESEARCH OF NEW YORK

Need some help understanding your Microdata™ computer? Then you need Osrow's Reality® Technical Journal, from Information Research of New York. The Journal's pages are packed with detailed explanations, discussions and examples showing how to use and operate Microdata computers. Written in an easy to understand tutorial format, the Journal has been read and applied by Microdata users everywhere. Whether you are a novice or an expert, the Journal is the perfect complement to confusing and opaque reference manuals.

SECTION I

Security: Protecting an Account, System Level Privileges, Monitoring System Usage with the ACC File, Protecting Program Files and Source Code, Security Verbs, Protecting a File, Locking Up Your System, Protecting an Attribute, Protecting a Terminal, Reserving a Terminal, Program Protection with Table Lookup • Learning the ENGLISH® Language, Dictionary Synonyms, Sorting Problems, The F Stack • System Efficiency: File Reallocations, The Restore Method, The Tape Method, The Copy Method • Debugging Procs • Converting Dates • Editor: Commands, Dealing with Lines, Multiple Prestore Commands • Reading an Item and Extracting an Attribute, Program and System Locks, The Program Debugger.

SECTION II

Creating User Accounts: Account Creation Schemes, Defining the New Account, Workspace Assignment, Q-Pointers, Proc Pointers • Cleaning Up Accounts and Removing: Dangerous Verbs/Procs, the Editor Ability, the ENGLISH Ability, the Bisynd, SCREENPROTM and Tape Verbs, the Spooler and Programming Verbs • Creating a Low Priority Account Creation Proc, Synonym Accounts • A Proc Primer • Security: Backing Up on Tape and Paper, File-Saves, Account-Saves, Dumping Files, Selective Restores • Dealing with Data: the Different Ways Data is Represented • Macros • ENGLISH: Creating Labels, F-Stacks • The RUNOFFTM Fill, Nofill and Justify Commands • Editor: Prestore Looping • Replacing Attributes and Writing Items, The READV and WRITEV Commands, The MATREAD and MATWRITE Commands, The Program Debugger.

SECTION III

Menu Construction • System Utilities: The Spooler, The SP-STATUS Verb, Assigning the Printer, Creating Hold Files, Printing Hold Files, A Proc to Print Hold Files, A Proc to Delete Hold Files • Proc: Transfers, Pointers, Jobstream Procs, A Proc Primer, The O Command, Screen Formatting, The X Command, The D Command • ENGLISH: The SELECT and SSELECT Verbs • List Management: Saving Items in a List.

SECTION IV

A Complete Program to Find the Report Length of an ENGLISH Proc: Introduction, Source, Listing, Narrative • Security: Controlling Electricity, Fire Protection, Protection Against Water Damage, Controlling Static Electricity • ENGLISH: Default Dictionary Attributes for Automatic Listings • A Proc Primer • System Utilities: The Spooler, The SP-KILL Verb • Editor: F, FS, FI, FD and EX Commands • Efficiency: Scheduling Jobs at Night, The TIMESLICE Verb, The POVFF Verb.

Want to learn more about using your Microdata computer? Do you need simple explanations telling how to make your computer sit up and beg? Order Osrow's Journal today! Available exclusively by subscription for \$125 since 1981, you can use the coupon to order your Journal for only \$49!

Name _____

Company _____

Address _____

City _____ State/Province _____

Country _____ ZIP/Mail Code _____

Telephone _____

Please send me _____ Journal(s) at \$49 each. The Journal contains all four sections listed.

SUBTOTAL _____

Add \$4 per Journal for North America shipping. Outside of North America, add \$7.50 per Journal.

SHIPPING _____

New York residents, please add appropriate sales tax.

TAX _____

Make checks payable to Information Research of New York. Mail your check and this form to: Osrow's Journal, 207 Granada Drive, Aptos, CA 95003.

TOTAL _____

ALLOW 4 TO 6 WEEKS FOR DELIVERY

PRAGMA

Issue Number 5

August, 1983

Pragma is published quarterly by

Semaphore Corporation
207 Granada Drive
Aptos, California 95003

Entire contents copyright © 1983 by Semaphore Corporation. All rights reserved. No part of this journal may be reproduced, transmitted, transcribed, stored in a recording, retrieval or computer system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of Semaphore Corporation.

Semaphore Corporation offers no warranty, either expressed or implied, for any losses due to the use of any material published in Pragma.

Subscriptions are \$25 per issue in the USA, \$38 per issue to other countries by airmail. All payments must be in US dollars drawn on a US bank.

Address all subscription correspondence to the Pragma Circulation Manager, Semaphore Corporation. When writing, enclose your issue's mailing address label or the numbers from the upper right corner of your label.

Address changes should be sent at least four weeks in advance. Include your new address along with your issue's mailing address label.

All correspondence and material received will be considered for publication. Address all submittals and correspondence to the Pragma Editors, Semaphore Corporation. All letters to the Editors are welcome and as many as possible will be published in the Letters Department. The Editors reserve the right to edit all submittals.

Is there a service Semaphore Corporation can provide for you? Address all inquiries to Semaphore Corporation or telephone 408-688-9200.

MENTOR is a trademark of Applied Digital Data Systems • ALL, DATA/BASIC, ENGLISH, MICRODATA, PRISM, REALITY, REFLEX, ROYALE, RUNOFF, SCREENPRO, SEQUEL, WORDMATE are trademarks of Microdata Corp. • INFORMATION is a trademark of Prime Inc. • ULTIMATE is a trademark of Ultimate Corp.

Happy Birthday!

With this first anniversary issue, Pragma begins its second year of publication. The entire staff would like to sincerely thank all of the readers and advertisers who have supported us over the past twelve months. Without your interest and enthusiasm, Pragma would not be possible.

...

This Pragma contains the much awaited index to all previous four issues. To insure that all of our regular features could still be included, the number of pages in this issue has been expanded to 56 (up from our usual 48).

...

How many Pick-style systems are out in the field? Based on manufacturers' claims, assume 11,000 machines have been shipped. Assume 10% of that iron is no longer alive and well. Assume one third of the living machines are at sites that don't know what a Pick system is. Assume two thirds of the remaining machines have one user actively looking to purchase one copy of one product in the Pick marketplace. Assume 15% actually buy. Software vendors take note: that equates to no more than 660 possible sales. Does anyone have more promising statistics?

...

New entry #7000 on our mailing list and winner of the (now) customary box of See's chocolates: Steve Jones of the Crisis Clinic in Seattle.

— The Editors

Back Issues Are Available!

Pragma #1: ZIP Code File Design • A Program for Renumbering Statement Labels • Deriving Year-to-Date Counts • VANILLA: The Parts File • Moletron User Profile • More Memory, Fewer Reads • SYSMAP: A Cross-Reference System • How to Flag Missing Checks • 23 Wish List Items • Computing Modules with ENGLISH® • Building ENGLISH Headings with PROC • 4 Queries • Making Item Identifiers Hash Well • Tape Handling in the Computer Room • Patching the Exit Problem in 3.2 SCREENPRO™ • The Shell Game.

Pragma #2: A Modulo Setting Program • Black Box Formatting • TRIM.DELIM and PROFILE Utilities • SYSMAP: File Format Input • Galinski Hamburg User Profile • LOOP vs. LOCATE • 25 Wish List Items • ENGLISH: Jargon • PROC Conversions • VANILLA: Bill of Material Input • 14 Queries • The Trouble Tree • 2 Letters • A Switchbox for 32 Ports • Security and the DATA/BASIC™ Programmer • The Cookie Game • Some New Subscribers.

Pragma #3: Edit Aides • A Proc for Cross-Referencing Q-Pointers • SYSMAP: Remaining File Input • Rainbow Natural Foods User Profile • More BASIC Benchmark Comparisons • Justifying Ragged Output • 13 Wish List Items • Generating Monthly Column Headings • VANILLA: Purchasing • 5 Queries • ENGLISH: More Commands • Shared Site Checklist • Converting Paint to Programs • 3 Letters • Generating Blank Forms • Animal, a Game that Learns • More New Subscribers.

Pragma #4: Uncompiling, Part 1 • SYSMAP: Dicts and Procs • Pacific Valley Bank User Profile • Left vs. Right Justification Benchmarks • 4 Wishes • A Comparison of BASIC Implementations • More New Subscribers • An Undocumented Editor Capability • 3 Local User Groups • Avoiding Saved Lists with PQ-RESELECT • Self-Documenting Reports • A Query • A Survey of Hardware that Supports Pick-Style Software • Printer Trade-Offs • ENGLISH: Finding Files • A Letter • VANILLA: Purchase Order Entry • Swat!

utilities

Uncompiling, Part 2: Regenerating Source Code

The second in a series of articles devoted to uncompiler object code is presented. The original version of the uncompiler from Part 1 is modified to allow compilable BASIC source statements to be reconstructed from object code.

The original uncompiler presented in the last installment [*Uncompiling, Part 1, Pragma #4, May 1983, page 12*], was really an "unassembler", in that it did little more than identify in mnemonic form the object code for each BASIC source statement. In order to regenerate compilable source statements from object code, the uncompiler must recombine the various mnemonic fragments into valid BASIC syntactical structures. This is best accomplished by essentially emulating a stack machine and "executing" the various operators as they are en-

Utility programs have saved many a programmer from the less appealing aspects of software development and maintenance. Reformatting code, stripping files clean of control characters, converting data from one format to another — all are examples of tasks best delegated to a program and not a programmer.

Good programmers will collect a "toolbox" of utility software to use from time to time. If you have a useful utility, send it in for publication. A regular feature in *Pragma* will be this Utilities Department, where good software tools will be spotlighted.

countered in the object code, to cause BASIC statements to be reconstructed in the machine's stack. The second version of the uncompiler, beginning on page 8, operates precisely in that manner.

For example, operands in the object code like variable names or string constants are simply pushed onto a STACK variable, such as in lines 37 and 63 of the uncompiler. When an operator is encountered, it typically causes one or more STACK elements to be combined into an expression, which is returned to the STACK. For example, the plus sign operator (HEXCODE "A9") stacks such an expression beginning at line 131. When the "final" operator for a statement (such as the code for PRINT) has been found in the object code, the completed source code is output by subroutine 1100 at line 144.

The dialog on page 9 shows actual output generated by the uncompiler as a result of processing the object code from a nonsense test program. Only a portion of all possible opcodes is tested for by the CASEs in lines 33 to 123 of this version of the uncompiler, so not every piece of object code will be successfully uncompiled. The STACK is output by line 25 if an unsupported opcode is encountered, to help the user decide how to interpret the STACK so that new code can be added to the uncompiler to correctly interpret the new opcode. Also, note how the opcode file below has been modified in some instances (such as the parentheses after DATE and the blanks around AND), to help simplify the generation of source code.

The next installment will address the problem of handling statement labels and GO TOs. (P)

00 NULL	01 END OF LINE	03 LOAD ADDR	05 PUSH	06 GO TO	10 GO IF ZERO	11 GO IF NONZERO
12 FOR TEST	13 GOSUB	14 RETURN TO	20 PUSH	22 PUSH ADDR	24 POP	30 PUSH MAT ADDR
32 PUSH MAT ADDR	34 POP MAT	40 PUSH	62 LOCK	63 UNLOCK	66 MAT INDEX	67 MAT INDEX
68 MAT ASSIGN	69 MAT COPY	6E INPUT USING	70 INPUT	71 INPUT, _	72 INPUT,	73 INPUT:
74 INPUT, :	75 INPUT, : _	76 PRINT	77 PRINT	78 PRINT	79 PRINTER CNTRL	7A PRINT ON
7B PAGE	7C HEADING	7D FOOTING	7E TAB	7F PROMPT	80 READC	82 READ
83 READU	85 READV	86 READVU	88 WRITE	89 WRITEV	8A MATREAD	8B MATREADU
8C MATWRITE	8D SELECT	8E READNEXT	90 OPEN	91 DELETE	92 CLEARFILE	94 RELEASE
95 READT	96 WRITET	97 WEOF	98 REWIND	9C PROCREAD	9D PROCWRITE	9E SHARE
A0 AND	A1 NOT	A2 OR	A3 <	A4 <=	A5 =	A6 MATCHES
A8 PUSH 1	A9 +	AA -	AB *	AC /	AD 0-	AE :
AF PUSH 0?	B0 STORE INDIRECT	B1 ON POP GO TO	B2 ON POP GOSUB	B3 SUBROUTINE	B4 CHAIN	B5 ENTER
B7 CALL	B8 CALL INDIRECT	B9 BREAK KEY ON	BA BREAK KEY OFF	BB DATA	BC ROM	BD CLEAR
BF RETURN	C0 ERRMSG ID	C1 STOP	C2 MARK ARGS	C3 DEBUG	C4 DELETE2	C5 EXTRACT2
C6 DELETE1	C7 EXTRACT1	C8 INSERT2	C9 REPLACE2	CA INSERT1	CB REPLACE1	DO DATE()
D1 TIME()	D2 TIMEDATE()	D3 ICONV	D4 OCONV	D5 NUM	D6 DELETE3	D7 EXTRACT3
D8 INSERT3	D9 LOCATE ATR	DA REPLACE3	DB ABS	DC INT	DE MOD	DF PWR
E0 RND	E1 SGRT	E2 LN	EB EXP	E4 COS	E5 SIN	E6 TAN
E7 @	E8 ASCII	E9 CHAR	EA COL1	EB COL2	EC COUNT	ED EBCDIC
EE FIELD	EF	F0 INDEX	F1 LEN	F2 SEQ	F3 SPACE	F4 STR
F5 []	F6 TRIM	F8 LOCATE SUBVAL	F9 LOCATE VALUE			

AUROPLANTM

AUROTECH
OF COLORADO

Now Software Similar To VisiCalc[®] Is Available On The Pick Operating System

A financial planning and modeling system that surpasses all others. Ideal for preparing financial models, budgets, cost estimates, sales recaps, scheduling and resource reports and analyses.

AUROPLAN is an electronic worksheet designed to drastically reduce the time and effort required to prepare financial reports.

AUROPLAN has a trial value feature which allows you to ask "What if?" without destroying previously entered data or formulas.

AUROPLAN allows you to name elements in one worksheet and reference them in another, so you can construct modular worksheets and link them together. This eliminates the need for huge worksheets and the long calculation times associated with them.

AUROPLAN can retrieve data directly from your existing

computer data base without the need for costly interface programming.

AUROPLAN can be learned in just a few hours. The only limit to its capability is your imagination.

That's AUROPLAN! Now available for use with the Pick Operating System.

For more information on
AUROPLAN, contact:

AUROTECH
OF COLORADO

925 S. Niagara
Denver, CO 80224
(303) 388-1612

VisiCalc[®] is a trademark of VisiCorp, Inc.


```

UNCOMPILER
001 EOL TO CHAR(1)
002 PRINT "FILE:"; INPUT FILE
003 OPEN FILE ELSE STOP "NO SUCH FILE!"
004 PRINT "ITEM:"; INPUT ITEM, I0
005 READ ITEM FROM ITEM, I0 ELSE STOP "NO SUCH ITEM!"
006 OPEN "OPCODES" ELSE STOP "NO OPCODES!"
007 DEL ITEM(1); FIRST.LINE = ITEM(1)
008 PRECISE = CHAR(SEQ(FIRST.LINE(5,1))+48)
009 IF PRECISE # 4 THEN PRINT "PRECISION":PRECISE
010 ITEM(1) = FIRST.LINE(6,LEN(FIRST.LINE)-5)
011 TOTAL.LINES = COUNT(ITEM,EOL)
012 IF TOTAL.LINES = 0 THEN TOTAL.LINES = 1
013 FOR I = 1 TO TOTAL.LINES
014 INSTRUCTS=FIELD(ITEM,EOL,I)
015 PTR = 1
016 STACK = ""; UNKNOWN = 0; ALREADY.PRINTED = 0
017 LOOP
018 OPCODE=INSTRUCTS(PTR,1)
019 UNTIL OPCODE="" DO
020 HEXCODE=CONV(OPCODE,"HX")
021 READ SOURCE FROM HEXCODE ELSE SOURCE = "OPCODE":HEXCODE
022 PTR = PTR+1
023 GOSUB 100; DECODE
024 REPEAT
025 IF UNKNOWN THEN PRINT "STACK=":STACK ELSE
026 IF ALREADY.PRINTED THEN PRINT "STACK=":STACK
027 END
028 NEXT I
029 PRINT "END"
030 STOP
031 *
032 100 * DECODE
033 BEGIN CASE
034 CASE (HEXCODE="03") OR (HEXCODE="20") OR (HEXCODE="22")
035 ADDR = INSTRUCTS(PTR,4)
036 PTR = PTR+4
037 INS ("V":ICNV(ADDR,"H00")) BEFORE STACK(1)
038 CASE HEXCODE = "24"
039 ADDR = INSTRUCTS(PTR,4)
040 PTR = PTR+4
041 OUTPUT = "V":ICNV(ADDR,"H00");" = ":STACK(1)
042 GOSUB 1100; * OUTPUT STATEMENT
043 CASE (HEXCODE="05") OR (HEXCODE="55")
044 OPERAND = ""
045 LOOP
046 NEXT.BYTE = INSTRUCTS(PTR,1)
047 PTR = PTR+1
048 UNTIL NEXT.BYTE = CHAR(254) DO
049 OPERAND = OPERAND:NEXT.BYTE
050 REPEAT

```

```

051 INS OPERAND BEFORE STACK(1)
052 CASE HEXCODE = "40"
053 OPERAND = ""
054 LOOP
055 NEXT.BYTE = INSTRUCTS(PTR,1)
056 PTR = PTR+1
057 UNTIL NEXT.BYTE = CHAR(254) DO
058 OPERAND = OPERAND:NEXT.BYTE
059 REPEAT
060 IF INDEX(OPERAND,"",1) THEN
061 OPERAND = "":OPERAND:"/"
062 END ELSE OPERAND = "":OPERAND:"/"
063 INS OPERAND BEFORE STACK(1)
064 CASE HEXCODE="76"
065 OUTPUT = SOURCE:" ":STACK(1):""
066 GOSUB 1100; * OUTPUT STATEMENT
067 CASE (HEXCODE="77") OR (HEXCODE="7C")
068 OUTPUT = SOURCE:" ":STACK(1)
069 GOSUB 1100; * OUTPUT STATEMENT
070 CASE (HEXCODE="7D") OR (HEXCODE="7F")
071 OUTPUT = SOURCE:" ":STACK(1)
072 GOSUB 1100; * OUTPUT STATEMENT
073 CASE HEXCODE="78"
074 OUTPUT = SOURCE; GOSUB 1100
075 CASE ("A2"<=HEXCODE) AND (HEXCODE<="A6")
076 GOSUB 800; * BINARY OPERATOR
077 CASE ("A9"<=HEXCODE) AND (HEXCODE<="AE")
078 GOSUB 800; * BINARY OPERATOR
079 CASE (HEXCODE="C1") OR (HEXCODE="C3")
080 OUTPUT = SOURCE; GOSUB 1100
081 CASE HEXCODE = "C5"
082 EXPR = STACK(3):<":STACK(2):":STACK(1):>"
083 DEL STACK(1); DEL STACK(1); STACK(1) = EXPR
084 CASE HEXCODE = "C7"
085 EXPR = STACK(2):<":STACK(1):>"
086 DEL STACK(1); STACK(1) = EXPR
087 CASE ("D0"<=HEXCODE) AND (HEXCODE<="D2")
088 GOSUB 700; * FUNCTION OF NO ARGUMENTS
089 CASE (HEXCODE="D3") OR (HEXCODE="D4")
090 GOSUB 1000; * FUNCTION OF TWO ARGUMENTS
091 CASE HEXCODE = "D7"
092 EXPR = STACK(4):<":STACK(3):":STACK(2):":STACK(1):>"
093 DEL STACK(1); DEL STACK(1); DEL STACK(1); STACK(1)=EXPR
094 CASE (HEXCODE="DB") OR (HEXCODE="DC")
095 GOSUB 900; * FUNCTION OF ONE ARGUMENT
096 CASE ("E0"<=HEXCODE) AND (HEXCODE<="E6")
097 GOSUB 900; * FUNCTION OF ONE ARGUMENT
098 CASE ("E8"<=HEXCODE) AND (HEXCODE<="EB")
099 GOSUB 900; * FUNCTION OF ONE ARGUMENT
100 CASE (HEXCODE="A0") OR (HEXCODE="EF")
101 GOSUB 800; * BINARY OPERATOR

```

UNCOMPILER VERSION TWO


```

:ED BP SAMPLES
TOP
.L22
001 Y = 1 ; Z = (X+Y)*123
002 PRINT "THIS ISN'T ALL":
003 LOGIC = (A<B) & (D=E) OR NOT(Y)
004 CATS = DOG<1,T,I>:OCONV(X,"MDO")
005 HEADING "PAGE ":DATE()
006 FORMULA = SQRT(COS(X)) ; Y=Y*Y
007 PRINT FIRST "R#10" : SECOND
008 PRINT @(3,5):STR('XYZ',3)[2,5]
009 STOP
010 END
EOF 10
.EX
-EXIT-

:BASIC BP SAMPLES
*****
LINE 10 [B0] COMPILATION COMPLETED

:RUN BP UNCOMPILER
FILE?BP
ITEM?$SAMPLES
V14 = 1 ; V28 = ((V1+V14)*123)
PRINT "THIS ISN'T ALL":
V5 = ((V32<V3) AND (V46=V50)) OR NOT(V14))
V82 = (V64<1,V6,V78>:OCONV(V1,'MDO'))
HEADING ('PAGE ':DATE())
V8 = SQRT(COS(V1)) ; V14 = (V14*V14)
PRINT ((V96 'R#10'):V0)
PRINT @(3,5):STR('XYZ',3)[2,5]
STOP
*
END

```

A sample dialog showing the source code for a nonsense program before compilation, and the resulting source code regenerated from the object code and listed by the uncompiler.

```

102 CASE HEXCODE="A1"
103 GOSUB 900 ;* FUNCTION OF ONE ARGUMENT
104 CASE ("F1"<=HEXCODE) AND (HEXCODE<="F3")
105 GOSUB 900 ;* FUNCTION OF ONE ARGUMENT
106 CASE (HEXCODE="ED") OR (HEXCODE="F6")
107 GOSUB 900 ;* FUNCTION OF ONE ARGUMENT
108 CASE (HEXCODE="DE") OR (HEXCODE="DF")
109 GOSUB 1000 ;* FUNCTION OF TWO ARGUMENTS
110 CASE (HEXCODE="E7") OR (HEXCODE="EC")
111 GOSUB 1000 ;* FUNCTION OF TWO ARGUMENTS
112 CASE HEXCODE="F4"
113 GOSUB 1000 ;* FUNCTION OF TWO ARGUMENTS
114 CASE HEXCODE="F5"
115 EXPR = STACK<3>:"[";STACK<2>:"";STACK<1>:"]"
116 DEL STACK<1> ; DEL STACK<1> ; STACK<1> = EXPR
117 CASE (HEXCODE="EE") OR (HEXCODE="F0")
118 EXPR = SOURCE:"(";STACK<3>:"";STACK<2>:"";STACK<1>:")"
119 DEL STACK<1> ; DEL STACK<1> ; STACK<3> = EXPR
120 CASE 1
121 UNKNOWN = 1
122 INS SOURCE BEFORE STACK<1>
123 END CASE
124 RETURN
125 *
126 700 * FUNCTION OF NO ARGUMENTS
127 INS SOURCE BEFORE STACK<1>
128 RETURN
129 *
130 800 * BINARY OPERATOR
131 EXPR = "(" ; STACK<2> : SOURCE : STACK<1> : ")"
132 DEL STACK<1> ; STACK<1> = EXPR
133 RETURN
134 *
135 900 * FUNCTION OF ONE ARGUMENT
136 STACK<1> = SOURCE:"(";STACK<1>:")"
137 RETURN
138 *
139 1000 * FUNCTION OF TWO ARGUMENTS
140 EXPR = SOURCE:"(";STACK<2>:"";STACK<1>:")"
141 DEL STACK<1> ; STACK<1> = EXPR
142 RETURN
143 *
144 1100 * OUTPUT STATEMENT
145 IF ALREADY.PRINTED THEN PRINT " ; " :
146 PRINT OUTPUT:
147 ALREADY.PRINTED = 1
148 RETURN
149 *
150 END

```

UNCOMPILER LISTING CONTINUED

producer profile

Interactive Systems is a small company located in Phoenix that develops and markets software for Pick operating system machines. Their Compu-sheet product is perhaps the largest selling independent software product in the Pick market. For this issue's profile, Pragma interviewed John Brandon, owner of Interactive Systems.

Pragma: How many Compu-sheets have been sold?

Brandon: Right now there are around 250 to 260 of them out there. It started off slow. I think the first four months or so I only sold maybe forty of them. Apparently more and more people are becoming aware of the product. In June I think I did forty in that one month. I perceive that as long as I can keep doing advertising, doing mailings, trying to reach that segment of the user base that I haven't already hit, I can see those sales continuing.

Pragma: What is the history of Compu-sheet?

Brandon: It goes back about two years ago, when Visicalc first came on the market and became so popular. My brother has an Apple, and one day I was over for dinner and he started telling me about Visicalc and how it worked. I was quite intrigued with the whole concept of the electronic spreadsheet. I thought that if we could take those basic concepts and apply them to the Pick operating system environment, we could offer a very powerful tool to the Pick user base. I sat down at that time and tried to put together some ideas as to how I would approach it, and kept running into brick walls, at least for about the first six months. I didn't spend six months actually working on it, but from time to time I would try to figure out what approach I would take. Finally, about eighteen months ago, I came up with some ideas, and Compu-sheet began to evolve. Interestingly enough, I was part of the way into it when I received a letter from Computer Distributors (CDI), who at that time were looking for an electronic spreadsheet application for their Series/1 implementation, and asked if I had one or knew of anybody who had one. At that time I decided I'd put a little more effort into completing the project. This was about March of 1982, and I had a version running by July that I demonstrated to CDI. They liked the product and subsequently accepted it with some modification. Following CDI's acceptance, I began to approach other vendors with the concept of bundling it into their operating system as CDI had wanted to do. I found that there was quite a bit of interest on the part of the vendors. At the same time, I was putting together a program for marketing the product to the end user. The next vendor that was showing some interest was General Automation, and I worked with them for several months. They did an analysis, as did CDI, of a number of different

How are hardware and software products created? How are they sold and marketed? Why are some successful and others are not? What should be done to make the next product even better?

Regardless of the size of a company that is selling some product, often only one or two employees at the company will control the engineering and marketing of the item being sold. In this regular department, Pragma will be interviewing those personnel intimately involved with the creation of popular and well known products in order to reveal the who, what, when, where, why and how of organizations currently offering goods and services to Pragma readers.

spreadsheets. They also decided to bundle Compu-sheet in with all of their Zebra systems. Following that, Datamedia also agreed to take the product and offer it to their customers. However, they are not bundling it with each system as are CDI and General Automation. They are instead offering it as an extra cost option. Most recently, we're negotiating right now with a company called Bantam which is taking the Pertec system and marketing it under their name.

Pragma: What have been your experiences in trying to market the product to end users?

Brandon: The marketing effort that goes into a product like this is easy to underestimate. The first problem I ran into was getting a list of my potential customers. It seems as though the lists in the Pick market are very difficult to come by and are guarded with almost a secrecy that's difficult to penetrate. Pragma's list has been one that has been available and it's probably the most comprehensive. There are other lists, but I think that even if we took all the lists that are available, we still could only reach 50% of the market. That's the first problem that you run into. The second problem is that I'm primarily a data processing professional, I'm not a marketer. At least I'm not the marketer that I felt was necessary to take a product like this to the marketplace and do a good job. So I decided at the front end that I was going to take it to a professional agency to get their ideas and suggestions as to how to put together a marketing program.

Pragma: Why did you decide to program in BASIC and to sell source?

Brandon: The total Pick user base is not that large. Half of them or so are Microdata users, but there's still a good percentage of other systems out there. I decided that if I tried to offer the product in assembly language, I would wind up maintaining a different version for every Pick machine. I just felt that I couldn't do that and support the product the way I really wanted to. BASIC has its disadvantages in some areas, particularly speed, but the advantage over assembly code comes in the area of support. I only have to maintain really two versions of the product, one for Microdata and one for Pick, and the differences between the two are really not that great. Actually there's another version for the Prime Information system. Still, the three versions are very close together, and there's really not much difference in the source code itself. So, I decided I would write it in BASIC to make it easier on me as far as maintenance goes. I no longer have to worry about whether my programs will function every time a manufacturer comes out with a new operating system. Another definite plus for selling BASIC source code is that when I find a bug, or if I want to make an enhancement, it's easy for me to put in a letter the changes that are necessary. All you have to say is

MULTIDATA'S NEW FIXED ASSET MANAGEMENT SYSTEM

...Handles the Exceptions as well as the Rules...

Multidata Corporation offers the most comprehensive on-line systems available. Each system is ergonomically designed to be truly user-friendly. Comprehensive "help" messages are provided at every data-input location. User-definable formats for data entry and retrieval let the system conform to your current business structure.

The Fixed Asset Management System provides a total solution to this complex problem. A representative sample of the features are:

CONTROLLER

- Corporate Elections/Standards
- All Depreciation Methods & Conventions including ACRS
- Handles non-standard items, such as partial fiscal year, 13 Accounting periods, or 4-4-5
- Depreciation projections, by period and G/L account
- Over 30 reports, each having multiple selections

TAX ACCOUNTANT

- Book federal and multiple state calculations
- ITC Calculations
- Ability to amend prior year records
- Gain/(loss) calculations
- Reconciliation of book-to-tax
- Ability to "what if" depreciation methods on an asset or entire property group

PROPERTY MANAGER

- Asset description
 - Vendor/manufacturer
 - Serial/Model Number
 - Location/Assignment
 - Tag Number
- Asset acquisition/disposition
- User definable formats
- Multiple locations (state, county, city, building)

By using Multidata's full support program, you are protected against any future changes in the law obsoleting your system.

ANOTHER MULTIDATA PRODUCT...

The Agency Management Support System provides Insurance Agencies and Brokers with the latest on-line data base technology for:

- Automated Case Tracking
- Marketing/Sales Support
- Electronic Filing/Mail
- Data Base Analysis
- Financial Analysis

Increase your profitability through greater productivity. A General Information Manual and Demonstrations are available. We would like the opportunity to tell you more about this Information Management System.



1570 The Alameda, Suite 110 • San Jose, California 95126 • (408) 293-1801

"edit this program and modify line 312 from this to that, then file it and recompile the program" and you have the bug fixed, without creating 250 tapes and mailing them to all the customers. That can be very expensive. As a result, I tend to notify my customers of problems and give them the opportunity to implement enhancements to the product rather frequently, and the users appreciate that because the product is kept up to date. Not only that, but have you ever talked to a consultant that wouldn't recommend or almost insist that the client have access to source code for any purchased software product?

Pragma: Even so, aren't there problems with media and operating system releases that the manufacturers keep changing on you?

Brandon: Actually, I've never run into a problem because of an operating system release. In one case there was a minor problem prior to release 1.6 on the ADDS Mentor operating system. The INPUT command didn't function properly when specifying a length. You're right, there is a hassle when it comes to media. You've got Microdata, Pick and Prime. Then I've got 800 and 1600 BPI tapes that I have to maintain. Then, just recently, 1/4 inch cartridge drives have become available, and I have to supply that media as well. Unfortunately, the vendors haven't decided to make all the 1/4 inch drives compatible, so each of the drives requires a copy to be recorded by that particular machine.

Pragma: How has developing and selling software turned out to be different than you expected when you started?

Brandon: There have been a lot of differences from what it's turned out compared to what I expected. One of the most interesting things that has come out of this is I had anticipated that dealing with customers would be somewhat difficult. There are really a bunch of good people out there, who are very easy to work with and are willing to work with you. I anticipated a little more of a problem in that area than what I wound up with. Support is another area. I think I underestimated the amount of effort it would take to support the users for a product like this. I find that about 50% of my customers will call me once with a question about some phase of the operation of Compu-sheet. I would say that out of those, probably half of those would call me two times, and a very small percent call me more than two or three times, a very small percent. We put together an eight page brochure that describes a lot about how the product works, yet prospects will still call and ask questions that are spelled out in the brochure. That's okay. I think that some customers just want to make sure that you're real, make sure that somebody answers the phone and there is going to be somebody there to support them when they have a problem.

Pragma: How difficult was negotiating with the system vendors?

Brandon: Once you can get in the front door, once you can get to the decision maker or the individual that is responsible for doing the evaluation, you find that they are very easy to work with, very easy to talk to. I was quite surprised in that area. I thought that they would be very difficult to work with. It didn't turn out to be that way at all, at least once you can get to that decision maker. That's the problem, getting to the right individual. That is very difficult to do.

Pragma: If you were introducing a new product now, what would you do differently?

Brandon: I would do a lot more advertising right up front. Even though I did a fairly respectable job of promoting Compu-sheet to the existing user base, I think I could have done a better job in that area right up front. I think if I were to

approach it again, I would have invested more money at the front end in a better advertising campaign.

Pragma: Is the Pick operating system becoming a large enough force in the industry, so that software vendors can make a living off of it?

Brandon: The existing user base is not very big. In order to make a living at it, you've either got to sell a heck of a lot to capture a large percentage of that existing base, or ask a lot of money for the product. I feel that the Pick operating system is without question the best business operating system available in the marketplace. I see it expanding at a rapid rate right now, but I'm disappointed at the rate at which it is expanding. I don't think that the manufacturers are really putting effort into educating non-Pick users in the advantages of the operating system. I would really like to see a lot more done in that area. We should see more articles in technical trade journals about the operating system, why it is as good as it is.

Pragma: What does the operating system lack?

Brandon: Communication I think is the most serious problem. We really need the ability to share a data base among a number of different machines. We need to be able to offer communication at a lot higher level than what is currently available. Machine communication as well as communication to other types of devices. I think that we need a lot better word processor, in order to support the small business establishment the way we'd like to. Pick should be able to slip into small businesses and service both data processing and word processing, because too many small businesses don't have the luxury of being able to afford both. We should be able to provide that to them.

Pragma: If I'm a software developer, and I can bring home an IBM personal computer for a couple of thousand dollars and write a program for an existing market of hundreds of thousands of users, why should I look at the Pick market at all?

Brandon: Because there is a customer base there with Pick. There is money to be made with Pick. In the IBM PC marketplace, you've got every 14 year old kid out there who has a PC sitting in his bedroom coming up with the idea of marketing software. If you pick up any PC magazine, 90% of its weight is in advertisements for other software products. If you were to develop a product for the PC marketplace, I think that the cost of promoting it in the right way would require an awful lot of capital. In the Pick marketplace, you don't have many software vendors out there. The need for good quality software exists, and if you develop a good product, it's quite easy to market. You don't have to battle with 50,000 other software developers.

Pragma: What are your feelings on the pricing of software?

Brandon: I feel that software in the Pick market has been pretty much overpriced. It's really been necessary to charge a lot of money for software simply because of the small potential base of customers that you have. But that's changing. It's kind of tough to go out there and ask two or three thousand dollars for software on a Pick machine when they can go to a local computer store and see it advertised for \$395 running on an Apple machine. As users become more and more aware, they realize that they don't really have to pay such high prices for software, that you can get quality software at low prices. I think the days of charging extremely high prices for software are gone.

Pragma: What are the pros and cons of having dealers sell your product for you?

Brandon: The pros are that you have others answering the

questions that demand your time. That aspect of it makes it very nice. You can still derive income from your software products without as much demand placed on you. The disadvantage is I wind up getting dealers calling me on the phone and they're all excited about Compu-sheet and they talk about how many customers they have out there and they want to be a dealer for it and so on. So we work with them and have them complete our software representative agreement, and go through all the formal procedures, and all too often you just don't hear all that much from them again. I don't know if it just gets put on the back burner or the salesmen just don't understand that much about the product to promote it, or where the problem lies. The initial enthusiasm, compared to the results some of them have, is a little bit disappointing, to say the least. On the other hand, there are good dealers out there who understand the needs of the market. They do a good job of marketing software for software houses. But there aren't as many of them who do a good job as there are who talk about doing a good job.

Pragma: Do you perceive any differences between the U.S. and foreign markets?

Brandon: I think the foreign market is a lot more sophisticated than the U.S. marketplace. For example, someone in England may make a decision to buy a computer. They just take the whole process a lot more seriously, computer and software both. I don't think that, over there, computers are the status symbol that you find they are in the United States. Foreign users keep you on your toes a lot more. You can tell a lot by the type of question that they ask. Maybe because it costs so much to make a phone call from England, they don't call you unless they have a good one. But it just seems that the level of

question tends to be very, very good, and very well thought out before they call.

Pragma: What about the logistics of trying to reach and support foreign users?

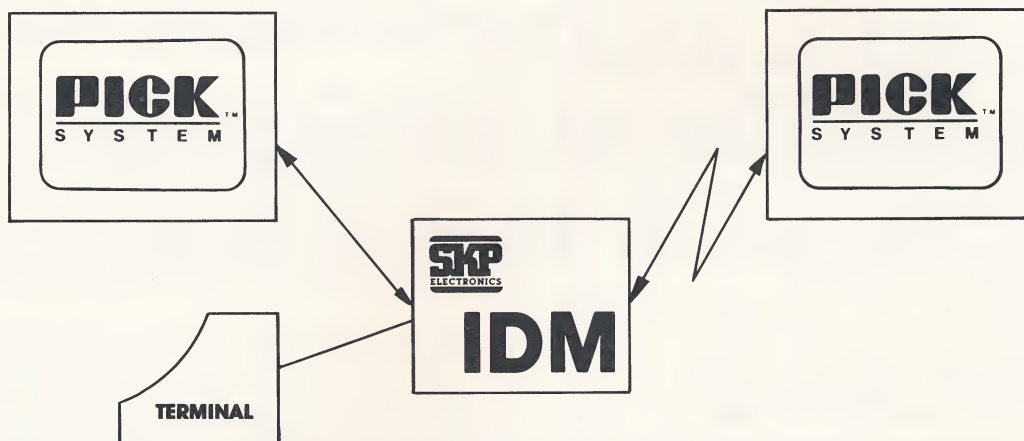
Brandon: It's much more expensive to reach that market. You can't do bulk mail, it costs you as much as regular mail. It costs four or five times as much to reach that market. The market isn't as big, and you don't get the same percentage response out of them as you do in the United States. I think justifiably so, because they must feel that if they needed support it would be much more expensive and much more difficult for them to receive it. I've often wondered whether or not the effort of trying to reach the overseas market has been justified. I can't say that it really has up to this point.

Pragma: What advice would you give the budding entrepreneur who is planning to unleash a software product?

Brandon: First of all, don't underestimate the scope of the project. I'm taking for granted that whoever is planning on doing this has done the proper market study and evaluation to determine the total potential customer base, and has done some basic calculations to determine whether or not it's worth getting into the project at all. Assuming that, I would say advertising is the most important portion of marketing the product. If you are not prepared to do it right the first time around, and get the concept and the idea of what you have to offer your customers across to them, you will definitely be disappointed in the performance of the product. You've got to be able to get out there and get the customer educated as to what you have to offer.

P

Networking With Pick Systems™



The Intelligent Data Mover (IDM) from S.K.P. Electronics will transfer items and/or files between **all** Pick System™ computers.



Contact:

Henry Herman
S.K.P. Electronics
1232-E South Village Way
Santa Ana, CA 92705
(714) 972-1727

Pick Systems is a trade mark of Pick Systems Inc.

A Day of Revelation

An evaluation of the Pick-style operating system called Revelation, available from Cosmos for the IBM personal computer, also known as the IBM PC.

Editor's note: Cosmos supplied Pragma with a complimentary copy of Revelation, knowing that we would be experimenting with the product and possibly doing an article reporting our findings. What follows is the one-day diary recorded by our staff as they tried the product for the first time.

10:00 AM: We open the large envelope delivered by UPS from Cosmos and discover the following contents:

- 1 packing slip claiming we have received "REVELATION W/CABLE".
- 1 345 page manual titled "INTRODUCTION, R/LIST, R/DESIGN".
- 1 280 page manual titled "R/BASIC".
- 1 pamphlet titled "HANDBOOK".
- 1 short ribbon cable with female connectors on both ends.
- 2 floppy disks, both with blue "R/BASIC" labels. "REVELATION" is typed on one label, "UTILITIES" is typed on the other.

The first thing we notice is that neither disk is write protected. Since we have just purchased a box of double sided, double density, soft sectored disks in anticipation of making backup copies of the Revelation disks, we immediately apply write protect tabs from that box of disks onto each of the Cosmos disks before we manage to accidentally clobber them. Cosmos should ship disks in permanently protected sleeves, like the disks you get when you buy a PC, or they should at least apply write protect tabs before shipping.

10:30 AM: We begin by skimming through the various manuals to get a feel for the system. The Handbook pamphlet consists of a few introductory notes, a software license, a warranty, and forms for reporting bugs and for designing applica-

tions software with the R/Design utility. The Handbook describes Revelation as consisting of:

- R/Basic, a procedural language
- R/List, a report generator
- R/Edit, a line editor
- R/Copy, moves files between Revelation and DOS
- R/File, creates, clears or deletes files
- R/Design, an application development system
- R/Net, a terminal emulator and download tool
- R/Upload, the complement of R/Net
- R/Buf, a printer buffer

That seems to be a lot more software than what was listed on the covers of the manuals. Do we really have all of the documentation? Yes, because it turns out that the two large manuals describe a lot more than their titles imply. The Introduction-R/List-R/Design manual actually contains 35 pages on introductory concepts (such as files and records), a 60 page tutorial on R/List, two more tutorials totaling 80 pages on R/Design, another 80 page reference section on R/Design, a 65 page reference section describing Revelation commands (such as CREATE-FILE and SORT), and three short appendices on file structures, R/Net and R/Upload. The R/Edit manual is found in the back of the R/Basic manual, which was briefly described in the article *A Comparison of BASIC Implementations* [Pragma #4, May 1983, page 18].

We notice that the Handbook pamphlet says our PC must include 320K of RAM, an 8087 math chip, and MS-DOS or PC-DOS (both are names for IBM's standard PC operating system), although it doesn't mention which release of DOS, while the Introduction manual doesn't say a thing about memory or chips but does mention that DOS 1.0, 1.1 or 2.0 all support Revelation.

11:00 AM: The Introduction shows that Revelation is invoked by first booting the machine under DOS and then inserting the Cosmos disk and giving the REV command to start executing Revelation, but first (in case something should go wrong) we begin making backup copies of the Cosmos disks using DOS's DISKCOPY and DISKCOMP commands.

11:15 AM: One of our new backup disks has started making horrible scraping noises while being copied. Apparently we didn't carefully center the disk while loading it and lowering the hub, which ended up clamping down on the disk instead of through the center hole. Now the edges of the disk around the hole are a bit wrinkled, which might have been avoided if our box of new disks had come supplied with hub rings. We notice the Cosmos disks already have hub rings. Good.

11:45 AM: We are done making backup copies. We have quickly come to the conclusion, as a result of giving DOS copy commands but not yet even having tried Revelation, that the IBM keyboard is very poorly designed and awkward to use. Apparently this has been a complaint of many PC owners. (This particular machine was borrowed just to let us try Revelation.) The SHIFT key is particularly poorly positioned and shaped.

Noon: With only a brief scan of the Introduction manual, and in just fifteen minutes, we have been able to successfully use many commands such as LIST, CREATE-FILE and COPY, and also to exercise much of the editor. A CUSTOMERS file already on disk (for the various documentation tutorials), and the similarity of so much of the system with other Pick machines, has made it particularly easy to jump right in and start exploring. Let's just keep poking around...

12:15 PM: More and more, we're becoming aware that the system sometimes momentarily pauses as the floppy disk

**Which method is your
company using to create
computer programs?**

**Method 1: WASTE
Method 2: WIZARD**

**Quit wasting time and
money. Use WIZARD and
start creating your
programs in minutes
instead of months.**

WIZARD is software that asks you "fill in the blank" questions about the kind of data entry program you want to create. WIZARD then uses your answers to automatically write the program for you.

To try WIZARD with no obligation, send your check for \$2100 to Automatic Programming. We will send you WIZARD on magnetic tape along with complete instructions. If within 30 days of receiving your tape you are not completely satisfied with WIZARD for any reason, we will permit you to return WIZARD, and we will then return your uncashed check to you.

Automatic Programming Inc. • 85 Lakeshore • Irvine CA 92714 • 714-552-2800


```

001 INDEX
002 EQUATE ALPHAS TO "ABCDEFGHIJKLMNORSTUVWXYXZabcdefghijklmnopqrstuvwxyz"
003 EQUATE DIGITS TO "0123456789.$"
004 *
005 OPEN "", "CSYM" TO OUT.FILE ELSE STOP "CSYM"
006 PRINT "File": INPUT FILE.NAME
007 OPEN "", FILE.NAME TO IN.FILE ELSE STOP FILE.NAME
008 PRINT "Item": INPUT ITEM.ID
009 READ ITEM FROM IN.FILE, ITEM.ID ELSE STOP ITEM.ID
010 EXECUTE "CLEAR-FILE CSYM"
011 LINE.NUMBER = 1 ; COLUMN = 1
012 LOOP
013 GOSUB GET.TOKEN:
014 UNTIL TOKEN = "" DO
015 READV FOUND.AT FROM OUT.FILE, TOKEN, 1 ELSE FOUND.AT = ""
016 FOUND.AT<1,-1> = LINE.NUMBER
017 WRITEV FOUND.AT ON OUT.FILE, TOKEN, 1
018 REPEAT
019 PRINT
020 STOP
021 *
022 GET.TOKEN:
023 GOSUB GET.BYTE:
024 TOKEN = BYTE
025 IF TOKEN = "" THEN RETURN
026 BEGIN CASE
027 CASE BYTE = CHAR(254)
028 LINE.NUMBER = LINE.NUMBER+1
029 PRINT " ":
030 GOTO GET.TOKEN:
031 CASE INDEX(ALPHAS, BYTE, 1)
032 LOOP
033 GOSUB GET.BYTE:
034 WHILE (INDEX(ALPHAS,BYTE,1) OR INDEX(DIGITS,BYTE,1)) AND (BYTE # "") DO
035 TOKEN = TOKEN + BYTE
036 REPEAT
037 GOSUB BYTE.BACK:
038 CASE (BYTE = ">") OR (BYTE = '<')
039 LOOP
040 GOSUB GET.BYTE:
041 UNTIL (BYTE = TOKEN) OR (BYTE = "") DO REPEAT
042 GOTO GET.TOKEN:
043 CASE BYTE MATCHES "1N"
044 LOOP
045 GOSUB GET.BYTE:
046 WHILE INDEX(DIGITS,BYTE,1) DO
047 TOKEN = TOKEN + BYTE
048 REPEAT
049 GOSUB BYTE.BACK:
050 CASE 1 ;# Ignore everything else
051 GOTO GET.TOKEN:
052 END CASE
053 RETURN
054 *

```

```

054 GET.BYTE:
055 BYTE = ITEMICOLUMN,1]
056 COLUMN = COLUMN+1
057 RETURN
058 *
059 BYTE.BACK:
060 COLUMN = COLUMN-1
061 RETURN
062 *
063 END

```

R/BASIC CROSS-REFERENCE GENERATOR

®

does a read. Even though we're seeing a wait of only a fraction of a second, we also know that only the speed of the floppy is what is holding things up. Already we're drooling for the XT model of the machine, with the ten megabyte hard disk.

This PC has a color monitor, and the convergence is not the greatest in the world, so each character (displayed as white on black) has a tiny rainbow-colored fringe. Also, as listings are output to the screen, or when keys are hit, the display "glitches" and small cursors seem to randomly appear on the screen. Why?

12:30 PM: Output to the printer seems to print with no problems, although buffering doesn't seem to be taking place since we don't get a colon prompt back until the printer is done. A quick search of the documentation fails to locate any information on R/Buf.

We notice that although the display waits after each page when doing a LIST, if we do a COPY of items to the screen, one item per page is displayed with no waiting between items, so the output just whips on by and we can't read it. There must be some kind of option or command to control that, but we can't find it.

12:45 PM: The awkward keyboard is getting to be a real nuisance. It's interesting how the hardware doesn't seem up to par with the software.

We have discovered a HELP command and a HELP file. The file mentions that R/Net will only work with 2.0 DOS. Let's see if we can get the PC talking to the Microdata™ in the next room.

1:20 PM: The terminal that happens to be sitting next to the PC is an ADDS Viewpoint on a 1200 baud line to the Microdata. We booted 2.0 DOS on the PC, started R/Net running, plugged the ribbon cable sent with Revelation into the serial port on the PC, plugged the 1200 baud line from the Microdata into the ribbon cable, and it all worked! We can now use the PC as a terminal on the Microdata.

We've noticed that R/Basic does not support an (X) option with the BASIC verb to allow cross-reference output. We've decided to write a cross-reference generator in order to see what writing a significant program with R/Basic is like. We'll create the program, which will be called INDEX, by modifying the old RENUMBER program from the first issue of *Pragma* [*RENUMBER: A Program for Renumbering Statement Labels, Pragma #1, August 1982, page 9*]. Since we already have a copy of RENUMBER on the Microdata, we'll download the source code to the PC to avoid having to retype the program and to exercise R/Net.

2:00 PM: The download is successful. With Revelation examples as a guide, we created and ran the following program

```
DOWNLOAD
001 OPEN "BP" ELSE STOP "BP"
002 PRINT "ITEM":
003 INPUT ID
004 READ ITEM FROM ID ELSE STOP ID
005 PRINT CHAR(2):ITEM:CHAR(3):
006 STOP
007 END
```

to allow the Microdata to read the RENUMBER item and transmit it to R/Net. In just a few seconds and after a handful of keystrokes on the PC, R/Net was done and had created a DOS file containing the item. One more COPY moved the item from a DOS into a Revelation file. Now we can edit

RENUMBER on the PC to turn it into INDEX.

4:40 PM: INDEX works, but it appears to run very slowly. (See listing on page 16.) Further tests will be needed to determine where the bottleneck is, but we'll have to leave that for some other time. Note that INDEX cross-references all symbols, including reserved words and numeric constants, even in comments. Reducing the output of INDEX to only variable names will be left as an exercise for the reader.

The printer on the PC needs a new ribbon, so let's see if we can use R/Upload to transmit INDEX back to the Microdata so it can be printed there.

5:15 PM: Again with Revelation examples as a guide, the following program

```
UPLOAD
001 DIM ITEM(100) : MAT ITEM = ""
002 OPEN "BP" ELSE STOP "BP"
003 PROMPT "e"
004 I = 1
005 LOOP
006 INPUT LINE
007 UNTIL LINE = CHAR(26) DO
008 ITEM(I) = LINE
009 I = I+1
010 REPEAT
011 MATWRITE ITEM ON "UPLOADED"
012 STOP
013 END
```

was written to allow the Microdata to receive text from the PC. R/Upload worked smoothly, and INDEX was successfully sent to the Microdata for printing.

5:30 PM: The day is over and our investigation of Revelation is done. The Revelation package is definitely powerful software. Although it is strictly a single-user system, it is familiar and easy to use for anyone with experience on Pick operating system machines. The reasonable price (\$950) and PC environment make it affordable and accessible. However, users should be careful that their PC has enough power (especially disk space and speed) to take advantage of the software.

In theory, remote workstations that offload processing chores from a central host computer are a good thing, but it remains to be seen just how well that approach works with PCs and a machine like a Microdata. (After all, downloading and uploading programs and data still doesn't give you a data base that is truly being simultaneously shared by two machines, while the cost of adding one more dumb terminal onto a fast enough host is very cheap today.) Revelation works fine as a stand alone system, but data bases often support multiple users, and there is only one keyboard on the PC. A local network of Revelation machines is probably the best future for this product. Unfortunately, R/Basic does not appear to currently support record locking, so it is not clear how Revelation programs would share files in a network.

The lack of an index in the Introduction manual was the only item that, for short moments, made Revelation a problem to use today. The product does deliver as advertised, and users who know how to fit microcomputers into the data processing scheme at their site will certainly get enough bang for the buck.

□

benchmarks

Are you thinking of doing an upgrade to your hardware or software? Are you comparing throughput and performance while shopping around for a system? Have you converted from one machine to another? Be sure to send in your benchmark statistics to Pragma, so the results can be featured in this department and shared with other installations.

IBM Personal Computer vs. Microdata™ Reality®

A variety of old and new benchmarks reveal large but unpredictable differences in the performance of these two machines.

The arrival at Pragma of the Revelation software [*A Day of Revelation*, page 14] has allowed us to make some new tests on an IBM PC with 384K of RAM and compare the results with what has almost become the standard ruler for this department: a Reality with 128K of MOS memory.

The table below contains the results of six different tests run on both machines. All times are in seconds.

The first two tests, LOOP and LOCATE, were originally described in this department last year [*Pragma #2, November 1982, page 20*] and included the times shown here for the Reality. The Revelation timings were obtained by Cosmos and were reported here earlier this year [*More BASIC Comparisons, Pragma #3, February 1983, page 25*]. The times equal a 14% improvement over Reality for LOOP on a PC, and 32% for LOCATE.

Pragma #3 also included the FIELD and OCONV benchmark programs and the Reality timings repeated here. The corresponding PC times (and all remaining table entries) were obtained during the *Day of Revelation*. In this case, Revelation can't keep up with Reality, slowing by 15% in the FIELD test and by almost a whopping 900% for OCONV! (The OCONV test does a FOR loop 10,000 times on Reality. Because Revelation took so long in this test, the loop was cut to 1,000 iterations and the PC then finished in 61 seconds, which was multiplied by ten to reach the figure in the table.)

The MATH test was the following program:

```
001 START = TIME()
002 FOR I = 1 TO 10 STEP .01
003 A = COS(I) ; B = SIN(I) ; C = TAN(I)
004 D = EXP(I) ; E = PWR(3,I) ; F = SQRT(I)
005 * PRINT I,A,B,C,D,E,F
006 NEXT I
007 PRINT (TIME()-START):" SECONDS"
008 STOP
009 END
```

The PC obviously makes good use of Intel's 8087 math chip (required when running Revelation), cutting 90% off the Reality time.

The final test was the compilation of a 352 line purchase order entry program consisting of 9,140 bytes of source code acceptable to the compilers on both machines. Revelation compiles remarkably fast, spewing out asterisks almost as quick as a Sequel, and over 70% faster than the Reality.

Of course, Revelation is limited to one user at a time, so comparing the Cosmos product with Reality is definitely comparing apples and oranges, but still it is fascinating to see just how the two compare.

P

	LOOP	LOCATE	FIELD	OCONV	MATH	COMPILE
Revelation:	133	32	45	610	10	33
Reality:	154	47	39	62	113	118

Automation Technology

TOTALWARE®

FEATURING TOTALLY INTEGRATED PACKAGES

Structured Software for Pick System® Hardware

MARKETWARE

A comprehensive package designed to provide several marketing tools to both managers and their sales staff.

SEARCHWARE

A personnel agency system designed specifically for permanent placement and consultant organizations.

PLANWARE

An electronic mail, time management and appointment scheduling system, which includes at a glance reporting.

MEDICALWARE

A medical practice system which dramatically reduces paperwork by automating third party insurance billing.

SALEWARE

A basic distributor package with functions from order entry inventory control to complete financials.

PHARMACYWARE

Allows pharmacists to fill prescriptions quickly and automatically generate third party insurance billing, resulting in increased payment time.

RENTALWARE

A property management system which encompasses all phases from tenant receivables to vacancy factor analysis.

**AN AUTHORIZED
GENERAL AUTOMATION
'ZEBRA' DEALER**

Call your local

(215) 885-4880

**TOTALWARE dealer or
Automation Technology today.**



**Automation
Technology**

873 Township Line Road, P.O. Box 7137, Elkins Park, PA 19117 • (215) 885-4880

VANILLA (The No-Frills Manufacturing System) Part 5: Receiving

This fifth installment in a series on the design and implementation of a manufacturing system presents the program for entering purchase order receipts.

The previous article in this series [*Pragma #4, page 32*] presented the GET.PO program for creating purchase orders. The RECEIVE program listed here on pages 22 and 23 is the

complement to GET.PO, in that it is designed to capture receipts against items in the PO file and to update the purchase order open balances to reflect the receipts that have occurred.

The file formats shown below include new descriptions of the IM file for tracking inventory quantities and the RC file of receivers, both of which are updated by the RECEIVE program. Also listed are some new attributes for the PO file that RECEIVE maintains. (See the last installment for descriptions of the other purchase order file attributes.)

The RECEIVE program is a fairly straight forward transaction processing program. Note that although no updates are made to the vendor file, VM items are still locked in the same manner and for the same reasons as described in the GET.PO documentation. Also, parts are considered destined for inventory if they are charged to an appropriate account number on the purchase order. RECEIVE arbitrarily assumes the correct inventory account number is 1234-56 (see line 102 of the program).

The next Vanilla installment will investigate parts inspection and quality assurance.

FILE FORMATS

File..	Attribute..	AMC *	Description.....
IM			Inventory master.
IM	PART	0	Part number.
IM	ON.HAND	1 A	Inventory quantity on hand. Does not include RI.QTY.
IM	RI.QTY	2 A	Quantity received and waiting to be inspected.
PO			Purchase orders.
PO	QTY.IN	20 S	Quantity received. This and RC are associated with the DATE.IN attribute.
PO	RC	21 S	Receiver number.
PO	QTY.RTV	22 V	Quantity of parts returned to vendor. Associated with the ITEM attribute.
RC			Receivers.
RC	NUM	0	Arbitrary receiver number assigned by receiving dock.
RC	PO	1 V	Purchase order numbers this receipt is for. ITEM and QTY are associated fields. QTYS from multiple inputs are merged so that an ITEM for a given PO never appears more than once.
RC	ITEM	2 V	Item number on purchase order this receipt is for.
RC	QTY	3 V	Quantity being received.
RC	ARRIVED	4 A	Date items arrived at receiving dock, as entered first time receiver was created.
RC	VENDOR	5 A	Number of vendor this receiver is from.

SCREEN DEFINITION

[illegible]

Which PICKTM-based electronic spreadsheet should you choose? COMPU-SHEETTM

You already know the value of an electronic spreadsheet. The question is which one should you choose? The answer is easy. COMPU-SHEET.

In the past nine months, over 250 end users have chosen COMPU-SHEET for their PICK-based computer systems. And more importantly, after months of extensive evaluation and comparison, CDI, General Automation and Datamedia have selected COMPU-SHEET as the best electronic spreadsheet for their customers. Those are pretty powerful endorsements, and the list is growing every day.

Why have so many people chosen COMPU-SHEET? Just compare COMPU-SHEET's key features with any other spreadsheet program:

- Very easy to use...no programming experience is needed.
- Uses simple English-like commands (COPY, MERGE, PRINT...)
- Worksheets may be merged or consolidated. The user controls the range of locations to be merged or consolidated.

- Formulas may specify retrieval of data from other worksheets.
- Formulas may specify retrieval of data from any file in the user's data base (down to the *sub-value level*, if desired).
- The user may enter a "?" at any point to get a "HELP" display.
- Multiple windows provide for viewing various sections of the worksheet at one time.
- The user has the option to control the format of any printed reports. Any combination of columns and rows may be selected.
- Formulas are written in simple algebraic format. Even "IF condition THEN formula1 ELSE formula2" is supported.
- A complete and formatted "AUDIT" report is available.
- Interfaces with "ACCU-PLOT" graphics system. Your worksheets may be represented as line graphs, pie charts, bar graphs, etc.
- Operates on Microdata, all PICK systems, and the Prime Information System.
- And more...

Plus, Interactive Systems offers you complete support after you choose COMPU-SHEET. If you

ever have a question, simply call us at (602) 993-3579, and we'll get you a helpful, friendly answer *fast*.

We want you to decide for yourself. Ask about our no-risk 21-day trial period. For information and a complete brochure, call Interactive Systems today or mail the form below. We're sure you'll choose COMPU-SHEET too.



INTERACTIVE SYSTEMS
129 East Voltaire Avenue
Phoenix, Arizona 85022
(602) 993-3579

**Yes! Please send me
more information about
COMPU-SHEET.**

Name _____
Company _____
Address _____
City _____
State _____ Zip _____
Phone () _____

COMPU-SHEET is a trademark of Interactive Systems.
PICK and PICK Systems are trademarks of PICK
Systems, Irvine, CA.

RECEIVE PROGRAM LISTING

```

001 EQU IM$ON.HAND TO 1
002 EQU IM$RI.QTY TO 2
003 *
004 EQU POS$VM.ID TO 1
005 EQU POS$QA TO 5
006 EQU POS$ITEM TO 7
007 EQU POS$PART TO 8
008 EQU POS$UM TO 9
009 EQU POS$DATE.IN TO 10
010 EQU POS$SCHED TO 11
011 EQU POS$QTY TO 12
012 EQU POS$BAL TO 13
013 EQU POS$DESC TO 15
014 EQU POS$VPART TO 16
015 EQU POS$ACCOUNT TO 17
016 EQU POS$QTY.IN TO 20
017 EQU POS$RC TO 21
018 EQU POS$QTY.RTV TO 22
019 *
020 EQU RC$PO TO 1
021 EQU RC$ITEM TO 2
022 EQU RC$QTY TO 3
023 EQU RC$ARRIVED TO 4
024 EQU RC$VM.ID TO 5
025 *
026 EQU VM$NAME TO 2
027 *
028 EQU item TO 2
029 EQU qty TO 3
030 EQU date TO 4
031 EQU receiver TO 5
032 EQU oldbal TO 6
033 EQU vnum TO 7
034 EQU vname TO 8
035 EQU part TO 9
036 EQU um TO 10
037 EQU desc TO 11
038 EQU vpart TO 12
039 EQU newbal TO 13
040 EQU command TO 14
041 *
042 OPEN "DF" TO DF.FILE ELSE STOP "VAN1"
043 OPEN "IM" TO IM.FILE ELSE STOP "VAN3"
044 OPEN "PO" TO PO.FILE ELSE STOP "VAN5"
045 OPEN "RC" TO RC.FILE ELSE STOP "VAN4"
046 OPEN "VM" TO VM.FILE ELSE STOP "VAN6"
047 READ SCREEN FROM DF.FILE, "RECEIVE" ELSE STOP "VAN7"
048 *
049 AT.ERR = @ (0,23):CHAR(27):"K"
050 *
051 LAST.PO = ""
052 10 *
053 PRINT CHAR(12):
054 100 *

055 RELEASE
056 PRINT @ (59,0):TIMEDATE():
057 INPUT PO.ID USING SCREEN, LAST.PO ELSE STOP
058 PO.ID = PO.ID<1>
059 *
060 READU PO FROM PO.FILE, PO.ID LOCKED
061 PRINT AT.ERR:"PO group locked!":
062 GO TO 100
063 END ELSE
064 PRINT AT.ERR:"PO not found!":
065 GO TO 100
066 END
067 LAST.PO = PO.ID
068 *
069 VM.ID = PO<POS$VM.ID>
070 INPUT IGNORE USING SCREEN, VM.ID AT vnum ELSE NULL
071 *
072 READU VM FROM VM.FILE, VM.ID LOCKED
073 PRINT AT.ERR:"Vendor group locked!":
074 GO TO 100
075 END ELSE
076 PRINT AT.ERR:"Vendor not found!":
077 GO TO 100
078 END
079 INPUT IGNORE USING SCREEN, VM<VM$NAME> AT vname ELSE NULL
080 110 *
081 INPUT ITEM USING SCREEN, "" AT item ELSE GO TO 10
082 ITEM = ITEM<1>
083 IF PO<POS$ITEM,ITEM> # ITEM THEN
084 PRINT AT.ERR:"Item not found!":
085 GO TO 110
086 END
087 *
088 TOT.SCHED = 0
089 LOOP UNTIL PO<POS$SCHED,ITEM,TOT.SCHED+1> = "" DO
090 TOT.SCHED = TOT.SCHED+1
091 REPEAT
092 IF TOT.SCHED < 1 THEN
093 PRINT AT.ERR:"Item has no scheduled deliveries!":
094 GO TO 110
095 END
096 *
097 PART = PO<POS$PART,ITEM>
098 INPUT IGNORE USING SCREEN, PART AT part ELSE NULL
099 INPUT IGNORE USING SCREEN, PO<POS$DESC,ITEM> AT desc ELSE NULL
100 INPUT IGNORE USING SCREEN, PO<POS$VPART,ITEM> AT vpart ELSE NULL
101 QA.PART = ((PART # "") AND (PO<POS$QA>="V"))
102 INV.PART = ((PART#"")) AND (PO<POS$ACCOUNT,ITEM> = "1234-56")
103 IF INV.PART OR QA.PART THEN
104 READU IM FROM IM.FILE, PART LOCKED
105 PRINT AT.ERR:"Part group locked!":
106 GO TO 110
107 END ELSE IM = ""
108 END
109 *

```


RECEIVE LISTING CONTINUED

```

110 INPUT IGNORE USING SCREEN, POKPOSUM, ITEM> AT um ELSE NULL
111 *
112 OPEN, BAL = 0
113 FOR J = 1 TO TOT, SCHED
114 OPEN, BAL = OPEN, BAL + POKPOSQTY, ITEM, J>
115 NEXT J
116 J = 1
117 LOOP UNTIL POKPOSDATE, IN, ITEM, J> = "" DO
118 OPEN, BAL = OPEN, BAL - POKPOSQTY, IN, ITEM, J>
119 J = J+1
120 REPEAT
121 OPEN, BAL = OPEN, BAL + POKPOSQTY, RTV, ITEM>
122 *
123 INPUT IGNORE USING SCREEN, OPEN, BAL AT oldba) ELSE NULL
124 *
125 1200 *
126 INPUT QTY, IN USING SCREEN, "" AT qty ELSE
127 IF INV, PART OR QA, PART THEN RELEASE IM, FILE, PART
128 GO TO 110
129 END
130 QTY, IN = QTY, IN<1>
131 INPUT IGNORE USING SCREEN, (OPEN, BAL-QTY, IN) AT newba) ELSE NULL
132 *
133 INPUT DATE, IN USING SCREEN, DATE() AT date ELSE GO TO 1200
134 DATE, IN = DATE, IN<1>
135 *
136 1300 *
137 INPUT RC, ID USING SCREEN, "" AT receiver ELSE
138 IF INV, PART OR QA, PART THEN RELEASE IM, FILE, PART
139 GO TO 110
140 END
141 RC, ID = RC, ID<1>
142 READU RC FROM RC, FILE, RC, ID LOCKED
143 PRINT AT, ERR: "Receiver group locked!":
144 GO TO 1300
145 END ELSE RC = ""
146 IF (RC<RC$VM, ID> # VM, ID) AND (RC# "") THEN
147 PRINT AT, ERR: "Duplicate receiver with different vendor!":
148 RELEASE RC, FILE, RC, ID
149 GO TO 1300
150 END
151 *
152 LOOP
153 INPUT COMMAND USING SCREEN, "" AT command ELSE GO TO 10
154 COMMAND = COMMAND<1>
155 UNTIL COMMAND = "F1" DO REPEAT
156 *
157 POKPOSDATE, IN, ITEM, -1> = DATE, IN
158 POKPOSQTY, IN, ITEM, -1> = QTY, IN
159 POKPOSRC, ITEM, -1> = RC, ID
160 *
161 IF QA, PART THEN
162 INKIM$RI, QTY> = INKIM$RI, QTY> + QTY, IN
163 END ELSE
164 IF INV, PART THEN INKIM$ON, HAND> = INKIM$ON, HAND> + QTY, IN

```

```

165 END
166 BREAK KEY OFF
167 IF INV, PART OR QA, PART THEN WRITE IM ON IM, FILE, PART
168 *
169 J = 1
170 IF RC = "" THEN
171 RC<RC$ARRIVED> = DATE, IN
172 RC<RC$VM, ID> = VM, ID
173 END ELSE
174 LOOP
175 RC, PO = RC<RC$PO, J>
176 UNTIL (RC, PO= "") OR ((RC<RC$ITEM, J>=ITEM)&(RC, PO=PO, ID)) DO
177 J = J+1
178 REPEAT
179 END
180 RC<RC$PO, J> = PO, ID
181 RC<RC$ITEM, J> = ITEM
182 RC<RC$QTY, J> = RC<RC$QTY, J> + QTY, IN
183 WRITE RC ON RC, FILE, RC, ID
184 *
185 * QTY, IN will shrink as it is distributed over open shipments
186 J = 1
187 LOOP
188 * Find next incomplete shipment, if any
189 LOOP UNTIL (POKPOS$BAL, ITEM, J> > 0) OR (POKPOS$BAL, ITEM, J>= "") DO
190 J = J+1
191 REPEAT
192 IF POKPOS$BAL, ITEM, J> # "" THEN ;# Open balance found
193 * Set difference to shipment's open balance,
194 * or to the amount left for closing shipments (QTY, IN),
195 * whichever is smaller
196 IF QTY, IN > POKPOS$BAL, ITEM, J> THEN
197 DELTA = POKPOS$BAL, ITEM, J>
198 END ELSE DELTA = QTY, IN
199 * Reduce shipment's open balance
200 POKPOS$BAL, ITEM, J> = POKPOS$BAL, ITEM, J> - DELTA
201 QTY, IN = QTY, IN - DELTA
202 END
203 WHILE (QTY, IN > 0) AND (POKPOS$BAL, ITEM, J> # "") DO
204 J = J+1
205 REPEAT
206 *
207 RELEASE VM, FILE, VM, ID
208 WRITE PO ON PO, FILE, PO, ID
209 BREAK KEY ON
210 GO 10
211 *
212 END

```


Two Microdata™ 8000 Systems Available

Each system equipped with:

512K MOS memory

(1) 128 MB Winchester Disk

(32) Communication Ports

(1) 1600 BPI Tape Drive

(1) 300 LPM Printronix Printer

\$59,000 (\$55,000 without printer)

(2) 5750 Communications Terminals also available

Contact: Jerry Haithcock
Electronic Engineering Co.
(216) 526-4350

PRICED TO SELL Complete Microdata™ system converted to Pick Evolution

64KB memory • 50MB disk
8 open ports • 800BPI tape
300LPM printer • 10MB Marathon disk
Isolation transformer

\$7,500

Contact John or Sheldon
S/C Performance Wholesale
1530 W. El Segundo Blvd.
Gardena, CA 90249
213-770-1100

FOR SALE Microdata™ Reality®

64KB Memory
50MB Disk
800BPI Tape
16 Ports
1 CRT Terminal
1 120CPS Printer

Call Jerry Buckley at 213-985-2680

FOR SALE Used Microdata™ Equipment

4 - Series 6000 800 BPI
tape drives
6 - Marathon 10 MB disk drives
10 - Prism® terminals
2 - Matrix printing terminals
1 - Matrix printer

Call 503-222-1977

wish list

Users of computer systems should always remember that the nice thing about software (besides the fact that it doesn't break when you drop it) is that programs are relatively easy to change — no soldering gun is necessary. So just because the operating system or the compiler or a system utility happens to work (or fail to work) in some particular fashion now, doesn't mean it has to always be that way. The next time an inspired idea arrives for improving your system, write it down and send it to Pragma for publication in the Wish List.

The previous 65 Wish List items have been featured in issues #1 through #4 of Pragma.

66. **Prompt Before Tape Tests.** Make the standard ACCOUNT-SAVE proc prompt for the tape label and account name *before* reading and writing the test patterns on the tape (which takes awhile when large records are being used), so that the operator doesn't have to sit and wait for the prompts while the tests are being done.

67. **Positioning S/NAME Attribute Headings.** S/NAME attribute headings in dictionary items are left justified and padded on the right with periods. Often these headings are more attractive when right justified or centered and padded with blanks or other characters. Any such custom heading can be constructed by putting the appropriate text in S/NAME, but the S/NAME text must always then be readjusted if the column width in V/MAX is changed. Provide a mechanism to allow centered and right justified attribute headings with arbitrary fill that automatically adjust to the current V/MAX, in the same way that left justified, period filled S/NAMEs currently adjust to V/MAX.

68. **Incremental FILE-SAVEs.** FILE-SAVEs of large disks can easily overflow one tape. This requires having the operator mount additional tapes, thereby slowing down the whole process and making it less automatic. Also, even though very few files may change between FILE-SAVEs, all files (unless they are defined with DX or DY pointers), get written to tape every time. Allow the system to optionally record which files were written to since the last FILE-SAVE. The FILE-SAVE command could then have an option to only save the files so marked. In that way, each FILE-SAVE would only save that portion of the system that changed since the last FILE-SAVE. Using a system of rotating tapes (say one tape for each day of the week), a nightly FILE-SAVE would only have to save the files that changed since the previous night, which might take very little tape and time. Once a week, a full save of all files could be done to restart the cycle and allow the tapes to be reused. There should also be a method to conveniently restore a system from such a set of tapes, which would be done by first restoring from the most recent complete FILE-SAVE tape, and then reading and overwriting the data from every remaining partial FILE-SAVE tape since then, in chronological order.

[P]

NEW FROM INTERACTIVE SYSTEMS

SCREEN-GEN



What is SCREEN-GEN?

SCREEN-GEN is a data entry or data inquiry processor that can be used to develop program applications such as: order entry, journal entry, inventory adjustment, customer inquiry, or any other screen-related process.

SCREEN-GEN has a language processor similar to ENGLISH or ACCESS that works with your data dictionaries in defining a screen program, its format, prompt and function fields, and the file update logic. Once generated, these programs can be run interpretively by the system, simply by keying in the program name.

SCREEN-GEN optionally allows you to generate a screen program into BASIC code. SCREEN-GEN generates comprehensive documentation on each program defined. It also includes a full screen editor for maintaining screen program definitions once they are created, and has a dictionary

editor for maintaining your data dictionaries.

SCREEN-GEN is easy to use, yet is comprehensive in its approach to solving data processing problems. Utilizing multi-valued data structures is no problem with SCREEN-GEN. You can define any number of windows in your screens that scroll forward or backward independently and allow full control of your multi-valued data.

With SCREEN-GEN, you can define programs that extend to multiple screens for data entry and update any number of files within your database. For customized processing, you can even include BASIC code statements and routines directly into your screen definitions that will allow you to accomplish in SCREEN-GEN what you might have achieved in a user-written program.

SCREEN-GEN is written entirely in BASIC; there are no assembly

language routines used and source

code is supplied to each user. You no longer need to be dependent on anyone else for the maintenance of your system. SCREEN-GEN is available for all PICK-compatible systems. **To learn more about SCREEN-GEN, call (602) 993-3579 or mail the form below to:**



INTERACTIVE SYSTEMS
129 East Voltaire Avenue
Phoenix, Arizona 85022

I'm interested in learning more about SCREEN-GEN.

Name _____
Company _____
Address _____
City _____
State _____ Zip _____
Phone () _____

SCREEN-GEN is a trademark of Interactive Systems. ENGLISH is a trademark of Microdata Corporation, Irvine, CA. ACCESS, PICK and PICK Systems are trademarks of PICK Systems, Irvine, CA.

SCREEN-GEN is to data entry what ENGLISH is to report generation.

MICRODATA®

USED EQUIPMENT & SYSTEMS

- 50 MB REFLEX I, DISC DRIVES
- 50 MB REFLEX I with CONTROLLER
- 128 MB REFLEX II, DISC DRIVES (NEW)
- 128 MB REFLEX II, with CONTROLLER
- 128 K MOS MEMORY BOARDS
- 16K CORE MEMORY BOARDS
- 8-WAY BOARDS with PORTS, CABLES
- INTELLIGENT 8-WAY BOARDS
- PRISM I CRTS
- PRISM II CRTS
- ADDS VIEWPOINT CRTS (NEW)
- 300LPM PRINTRONIX PRINTERS (NEW)
- PRINTER CONTROLLERS with CABLE
- 600LPM PRINTRONIX PRINTERS (NEW)
- CORE SYSTEMS
- MOS SYSTEMS

CALL CHUCK HAAS FOR
COMPETITIVE PRICES

BUY-TRADE-SELL

513-528-5400

PLEASE NOTE OUR NEW TELEPHONE NUMBER

EASTCOMP

II



Proven Pick System Software

for

- Property Management/Real Estate
- Oil & Gas
- Capital Equipment Leasing
- Staff Scheduling
- Distribution
- General Accounting

from

CTS Services, Inc.
214/637-3344

An Introduction to ENGLISH® Part 4: Being Choosy

The fourth in a series of articles is presented for the beginning user of the inquiry and report generation language called ENGLISH. Item lists are introduced to allow the selection of only certain items for output.

You may have noticed that whenever you LIST a file, the display screen will quickly fill up with data and the system will wait for you to hit the RETURN key before displaying the next page. One way to avoid having to page through long lists of items on the screen is to print the report instead of display it. If your computer (not your terminal) has a printer, then the output of any ENGLISH command you give can be printed at the printer instead of displayed at the terminal simply by adding the word LPTR to the end of the command.

Even when you can make printouts, it would still be nice to be able to stop the display of items once you have seen enough data in a report. This can be accomplished by hitting the BREAK key, which will cause an exclamation point (!) prompt to appear. Now you can type END and hit RETURN to cause the computer to immediately end the report being displayed. Usually you can hit BREAK at any time to stop what your terminal is doing. Occasionally you may get an asterisk (*) prompt instead of an exclamation point, but just treat it the same and enter END to stop the output.

Even with the ability to stop a report, it can still take forever to wade through long displays to find only the few items in a file that may be of interest. If you know the identifiers (remember the unique numbers?) of the items you want to display, then enclose the item identifiers in single quotes in your command, and the report will then consist of only those items. For example, if you want to know the ZIP code for customer M719, then you might use a command like LIST CUST 'M719' ZIP (assuming the customer file is called CUST and its dictionary has the word ZIP).

If you wanted to list all purchase orders numbered less than 200, having to type 199 item identifiers in a command would be unreasonable. Fortunately, you can use *relational operators* to denote ranges of items. The operator EQ means equal, GT means greater than, LT means less than, GE means greater than or equal to, and NE means not equal to. A command like LIST PO LT '200' would list only the purchase orders numbered less than 200.

The words AND and OR can also be used to connect relational operators into longer series of tests. For example, the command LIST SO GE '1000' AND LT '2000' might be used to list only those sales orders from #1000 up to but not including #2000. What does the command LIST VENDOR GT '10' AND LT '30' OR GT '40' AND LT '50' do? Can you tell what the command LIST ORDER NE '553' AND NE '729' AND NE '802' does?

P

**How Can
Your Direct
Mail Flyer
Reach
Thousands
of Pick
Users for
Only 34¢
Per Name?**

**To find out,
call Pragma today!
408-688-9200**

**LET MADIC HELP YOU CONTROL
YOUR MANUFACTURING BUSINESS**



MADIC 3960 Freedom Circle Santa Clara, CA 95054 (408) 988-8311

**Call Today
Or Send Us Your
Business Card
For More Information**

DOLARS™

Distributed On-Line Accounting and Reporting System

MAKES SENSE

The BEST Financial Software for the PICK Operating System

DELIBERATE SYSTEMS

**3456 Camino Del Rio North • San Diego, CA 92108
(619) 280-4400**

Converting Manual Paint to Programs

A previously published program that generates other programs is modified to demonstrate it can accept even manually designed input.

The article *Converting Paint to Programs* [Pragma #3, February 1983, page 34] input lines of text stored as subvalues in an item, and generated as output a BASIC program to recreate the text. Unfortunately, because the program assumed the Screenpro™ painter would be used to generate the input item, many readers ignored the CODE-PAINT generator and didn't realize it would work just as well with input manually created with the editor. In fact, by changing line 7 in the original program to use CHAR(254), and line 10 to ex-

tract PAINT's attribute instead of subvalue, the program would then accept text stored as attributes, just as humans prefer.

The new version of the program below incorporates those changes, and also generates slightly better code, in that the STR function is used whenever the string constant being output is a repetitive list of more than six characters. (Consult your compiler documentation to determine at exactly what length it becomes most efficient for your implementation to use STR.)

The dialog at right demonstrates how the PAINT.TO.CODE program is used. First, the user lists an item called FORM, which is nothing more than manually entered lines of text arranged in the form desired by the user. Next, PAINT.TO.CODE is executed and it outputs a new BASIC program that includes all the PRINT statements (41 in this example) necessary to generate the form. The name of the program is created by prefixing "BASIC." to the input item name. The new program is then compiled and RUN to verify it indeed outputs a duplicate of the manual form. The program can now be edited to turn it into a final production program that prints actual values on the form, but at least most of the mundane work of generating PRINT statements has already been automatically done. P

```
001 PRINT 'DATA FILE' :
002 INPUT FILE.NAME
003 OPEN FILE.NAME ELSE STOP 'NO SUCH FILE!'
004 PRINT 'ITEM' :
005 INPUT ID
006 READ PAINT FROM ID ELSE STOP 'NO PAINT!'
007 ROWS = COUNT(PAINT, CHAR(254))+1
008 CODE = ''
009 FOR LINE = 1 TO ROWS
010     TEXT = PAINT<LINE>
011     IF TEXT # '' THEN
012         LOOP
013             COL = 1
014             LOOP UNTIL TEXT[COL, 1] # ' ' DO COL = COL+1 REPEAT
015             CHARS = TEXT[1,COL-1]
016             GOSUB 100 ;* Generate PRINT
017             NON.BLANK.POS = COL
018             LOOP
019             COL = COL+1
020             DELIMITER = TEXT[COL, 2]
021             UNTIL (DELIMITER = ' ') OR (DELIMITER = '') DO REPEAT
022             CHARS = TEXT[NON.BLANK.POS, COL-NON.BLANK.POS]
023             GOSUB 100 ;* Generate PRINT
024             WHILE DELIMITER # '' DO
025                 TEXT = TEXT[COL, LEN(TEXT)-COL+1]
026             REPEAT
027             END
028             CODE<-1> = 'PRINT'
029         NEXT LINE
030     CODE<-1> = 'STOP'
031     CODE<-1> = 'END'
032     WRITE CODE ON 'BASIC.' : ID
033     STOP
034     *
035     100 * Generate PRINT
036     LENGTH = LEN(CHARS)
037     IF LENGTH <= 0 THEN RETURN
038     FIRST.CHAR = CHARS[1, 1]
039     IF (COUNT(CHARS, FIRST.CHAR) = LENGTH) AND (LENGTH > 6) THEN
040         CHARS = 'STR(' : FIRST.CHAR : ', ' : LENGTH : ')'
041     ELSE CHARS = ' ' : CHARS : ' '
042     CODE<-1> = 'PRINT ' : CHARS : ' : '
043     RETURN
044     *
045 END
```

PAINT.TO.CODE PROGRAM GENERATOR

ED BP FORM

TOP

.L22

001 +-----+

002 | ORDER NUMBER: |

003 +-----+

004

005 =====

006 ITEM | PART NUMBER | QTY | TOT COST

007 =====

008 | | |

009 =====

010 SUBTOTAL:

011 TAX:

012 +-----+

013 | TOTAL: |

014 +-----+

EOF 14

.FI

'FORM' FILED.

:RUN BP PAINT.TO.CODE

DATA FILE?BP

ITEM?FORM

:CT BP BASIC.FORM

BASIC.FORM

001 PRINT "+-----+" :

002 PRINT

003 PRINT " | ORDER NUMBER: " :

004 PRINT STR(" ", 19) :

005 PRINT " | " :

006 PRINT

007 PRINT "+-----+" :

008 PRINT

009 PRINT

010 PRINT STR("=", 35) :

011 PRINT

012 PRINT "ITEM | PART NUMBER | QTY | TOT COST" :

013 PRINT

014 PRINT STR("=", 35) :

015 PRINT

016 PRINT " " :

017 PRINT " | " :

018 PRINT STR(" ", 13) :

019 PRINT " | " :

020 PRINT " " :

021 PRINT " | " :

022 PRINT

023 PRINT STR("=", 35) :

024 PRINT

025 PRINT STR(" ", 17) :

026 PRINT "SUBTOTAL:" :

027 PRINT

028 PRINT STR(" ", 22) :

029 PRINT "TAX:" :

030 PRINT

031 PRINT STR(" ", 18) :

032 PRINT "+-----+" :

033 PRINT

034 PRINT STR(" ", 18) :

035 PRINT " | TOTAL: " :

036 PRINT STR(" ", 8) :

037 PRINT " | " :

038 PRINT

039 PRINT STR(" ", 18) :

040 PRINT "+-----+" :

041 PRINT

042 STOP

043 END

:BASIC BP BASIC.FORM

LINE 43 [B0] COMPILATION COMPLETED

:RUN BP BASIC.FORM

+-----+

| ORDER NUMBER: |

+-----+

=====

ITEM | PART NUMBER | QTY | TOT COST

=====

| | |

=====

SUBTOTAL:

TAX:

+-----+

| TOTAL: |

+-----+

:

Ⓟ

INVOKING PAINT.TO.CODE TO GENERATE A FORMS PROGRAM

local user groups

Where are the local user groups and what are they up to? This regular department is designed to help our readers stay aware and informed of nearby groups and activities for computer users.

Why isn't YOUR user group mentioned below? To keep us informed of your group's activities and to appear in this department in future issues, keep **Pragma** on your mailing list and send us your group's newsletter!

Toronto

The Canadian Pick Users (also known as CPU) is a group formed four years ago that now consists of approximately fifty members, representing Microdata™, ADDS, Prime, Ultimate and Evolution installations. The group meets six times a year to hear a guest speaker and to discuss a technical topic at each meeting. Six members form an Executive Board that meets once a month to decide when and where to hold meetings and who will be the guest speaker. The Annual General Meeting will be a luncheon September 14th at the Prince Hotel on York Mills Road in Don Mills, Ontario. That session will include the voting for a new slate of Executive members. Users interested in more information about the CPU should contact Secretary Marie Carter at Box 876, Postal Station B, Willowdale ON, M2K 2R1.

Washington DC

The National Capital Area Microdata Users Group meets once a month for dinner and a guest speaker presentation. The June meeting featured a talk on the psychology of user interaction and satisfaction with computer systems, by Kent Norman of the Psychology Department of the University of Maryland. The July meeting included Monisa Trice of Drs. Groover, Christie and Merritt, who shared her experiences regarding the selection process that led to the purchase of a Sequel™. That meeting also featured a description of the Sequel from a hardware point of view by Todd Smoot, a former Microdata customer engineer, and comparisons of the Sequel with Reality, Evolution, Ultimate, and other Pick machines. Interested users should contact Stan Seidlitz, Continental Computer Services, 9489 Silver King Court, Fairfax, VA 22031 or telephone 703-273-0816.

Northern California

The Northern California Pick Users (NCPU) has been a busy group during the past quarter, having met in May for a presentation by Cliff Diehl of Systems Management on the installation and use of MRP systems, followed by the June meeting where Roger Harpel of Cosmos gave a demonstration of the Revelation system for the IBM PC, and most recently at a gathering in July that featured a presentation on forms management. The NCPU newsletter is consistently the longest and most newsy user group publication received at **Pragma**. The May issue described the claims of one member that there are no less than seventeen different Pick machine implementations (twelve "official" and five others counted but not described). The August meeting is scheduled to be held in San Francisco and will feature a presentation on stress

management, while the September meeting will have Bill Walsh of Pick Systems as guest speaker. Contact Secretary Lisa Levsen for more information by writing Box 6759, Oakland, CA 94603 or calling 415-632-0977.

Southern California

The California Data Base Management Association (CDBMA) rated its June outdoor meeting and dinner dance a "smashing success". The gathering, held at San Diego State University, featured a program on application generators by Dick Schmoekel of the Software Express. The barbecue steak dinner and drinks were free to all new members. The July meeting in Irvine featured a presentation on managing careers, and also included a technical session by Alan Thompson of Hudson Oxygen on string manipulation using dynamic arrays. The annual Member of the Year Award will be presented at the August meeting in Newport Beach. Linda Ristow is the Secretary of this group, which can be reached by writing CDBMA, 9740 Appaloosa Road #210, San Diego, CA 92131 or by calling 619-578-3152.

Arizona

Our most recent letter from the Pick Users Group of Arizona reports on the April meeting which featured guest speakers Roger Brown, a Microdata region manager, and two of his co-workers, Fred Crow and Gary Monroe. Brown discussed the structure of Microdata's organization and the new divisions and developments of the past year. Technical questions were fielded by Crow and Monroe. The April newsletter included a copy of a Microdata memo responding to questions about any Microdata plans to offer power conditioning equipment, and on the reliability of some diagnostic programs, which couldn't be answered at the meeting. For more information, contact Secretary Jodi Hilgenberg, Communication Skill Builders, Box 42050, Tucson, AZ 85733.

Delaware Valley

The May meeting of the Delaware Valley Data Base Management Association featured a presentation on the ADDS Mentor by Judd Van Dervort of Keystone Information Systems and a discussion on system security led by Jim Carroll and Karl Carapellotti. The last meeting for the summer season was held in June and included panel discussions on user developed software, by George McGann and Bob Anderson, and on efficient programming, by Jeff White and Steve Love. For information on becoming a member when meetings resume, contact President Jim Cates, Pars Manufacturing, Box 149, Ambler, PA 19002. P

PRODUCTIVITY;

How is it measured by manufacturers?
The MRP Software industry measures it as:

CAPABILITY

PRICE

\$9995

PRO:MAN, PROductivity for MANufacturing is an integrated, on-line, net change, closed-loop MRP II system with complete financials available on PICK/OS computers (IBM, DEC, Honeywell, Microdata, Altos, etc.)

R.S.D. Inc.
13333 N.E. Bel-Red Road
Bellevue, WA 98004
(206) 451-0404

Interested?

Contact:
Pat Carey

Sales Order Processing
Sales Analysis
Invoicing

Accounts Receivable
Accounts Payable
Payroll
Cost
General Ledger
Budgeting
Fixed Assets

Bills of Material
Master Scheduling
Inventory Transactions
MRP
Scheduled Receipts
Capacity Planning/
Routing

Purchase Order Control
Shop Floor Control

Some New Subscribers

David Carlson
PIA of New England
Hopkinton, MA

Henry Stone
Connecticut Underwriters Inc.
Portland, CT

Kenneth R. Davis
Copper Cellar Corp.
Knoxville, TN

Don Oltmans
Oltmans Construction Co.
Monterey Park, CA

Karen Stine
Oryx Press
Phoenix, AZ

Gerald A. Saunders
Gerald Saunders Systems Ltd.
Richmond, VA



National Customer Engineering, Inc.
Computer Maintenance and Consulting

COMPUTER MAINTENANCE SPECIALISTS

The finest service money can buy.

619-435-8363

NATIONAL CUSTOMER ENGINEERING, INC.

P.O. BOX 1260

CORONADO, CALIF. 92118

A Program that Reports File Pointer Locations

A simple but convenient utility program for reporting the location and equivalence of file dictionary and data pointers is presented.

During software development and maintenance, a programmer will frequently be logged onto an account and be thinking of a name that supposedly identifies a file and will suddenly want to know the answer to one or more of the following related questions:

1. Is the name in the Master Dictionary?
2. If so, is the item the name of a file?
3. If so, is the file actually defined in this account (via a D-pointer item, as opposed to a Q-pointer that acts as a synonym for a file actually defined in some other account)?
4. If so, does the file have both a dictionary and a DL/ID pointer?
5. If so, does DL/ID point to and equal the dictionary?
6. In any case, what are the base, modulo and separation in the file's D-pointers?

The program below will answer all of the above questions in one simple step either by successfully locating the necessary D-pointers and reporting the base, modulo and separation for the file, or by reporting failure instead. The proc

```
LOC.FILE
001 PCN
002 HRUN BP LOC.FILE
003 P
```

should be defined in the Master Dictionary so that the program can be invoked as a verb by entering LOC.FILE followed by the suspected file name.

Ⓟ

```
001 OPEN "MD" ELSE
002   PRINT "MD not found!"
003   STOP
004   END
005 PROCREAD COMMAND ELSE
006   PRINT "Use LOC.FILE command!"
007   STOP
008   END
009 FILE = COMMAND<2>
010 READ DITEM FROM FILE ELSE DITEM = ""
011 IF DITEM<1> = "Q" THEN
012   PRINT FILE:" is a Q-pointer!"
013   STOP
014   END
015 OPEN "DICT", FILE ELSE
016   PRINT FILE:" is not a file!"
017   STOP
018   END
019 BASE1 = DITEM<2>
020 MOD1  = DITEM<3>
021 SEP1  = DITEM<4>
022 READ DITEM FROM "DL/ID" ELSE DITEM = ""
023 BASE2 = DITEM<2>
024 MOD2  = DITEM<3>
025 SEP2  = DITEM<4>
026 IF (BASE1=BASE2) AND (MOD1=MOD2) AND (SEP1=SEP2) THEN
027   PRINT FILE:" DICT @ ":BASE1:" x ":MOD1:" x ":SEP1
028   END ELSE
029     PRINT FILE:" DICT @ ":BASE1:" x ":MOD1:" x ":SEP1
030     IF BASE2 # "" THEN
031       PRINT FILE:" DATA @ ":BASE2:" x ":MOD2:" x ":SEP2
032       END
033     END
034 STOP
035 END
```

LOC.FILE PROGRAM LISTING

Ⓟ


```
:CT MD PRINT.MEMOS
```

```
PRINT.MEMOS
001 PQN
002 HSELECT EMPLOYEES
003 STON
004 HPQ-SELECT 1
005 P
006 100 MV &1 !1
007 IF # &1 X
008 FB (EMPLOYEES &1)
009 GO 100
010 HRUNOFF DOC MEMO
011 STON
012 MV #1 &1
013 H<
014 P
015 GO 100
```

```
:CT EMPLOYEES *
```

```
498
001 SUSAN WATERS
002 PAYROLL
```

```
726
001 LESLIE FORTUNA
002 ORDER ENTRY
```

```
:CT DOC MEMO
```

```
MEMO
001 .LINE LENGTH 30
002 .SK 2
003 October 10, 1983
004
005 To:
006 .INPUT
007
008 The location of the annual
009 company Halloween party has
010 been changed to...
```

```
:PRINT.MEMOS
```

```
2 ITEMS SELECTED.
```

```
October 10, 1983
```

```
To: SUSAN WATERS
```

```
The location of the
annual company Halloween party
has been changed to...
```

```
October 10, 1983
```

```
To: LESLIE FORTUNA
```

```
The location of the
annual company Halloween party
has been changed to...
```

```
:
```

Ⓟ

command files

Boiler Plate Processing with Runoff™

Customized copies of a Runoff document can be created even when .READNEXT commands are unavailable.

Some versions of Runoff include a .READNEXT command specifically designed to allow custom text to be inserted into document items, while other versions of Runoff, such as on Microdata™ machines, do not support a .READNEXT command. But Microdata computers can still generate multiple copies of a Runoff document, with each copy containing a mixture of unique text, through the use of .INPUT commands and careful proc programming.

The dialog at left shows how one copy of a standard memo can be output any number of times, with each copy of the memo uniquely addressed to a different employee name. In this example, the user first lists the PRINT.MEMOS proc and sees that it SELECTs all items in an EMPLOYEES file and does a RUNOFF of an item called MEMO in the DOC file for each employee. The user then inspects the EMPLOYEES file and finds two items (although there could be any number of items) with the employee's name in attribute one of each item. Inspection of the MEMO item reveals an .INPUT command imbedded in the text at the point where the addressee's name is desired. Finally, the user executes the PRINT.MEMOS proc and sees that one copy is output and addressed to each employee.

The proc works by MVing (at line 12) the first attribute of each SELECTed item into the secondary output buffer so that the .INPUT command will take the data from there instead of prompting the operator. Note that the MV could just as easily work with any attribute number, and that any number of MVs (along with a corresponding number of .INPUTs) could be used, allowing each output document to have any number of custom lines, such as a complete address with a street, city, state and ZIP to appear with each name.

Ⓟ

TRY IT, YOU'LL LIKE IT

It's not easy selling software. First you write it, make sure it all works correctly, document it to a fare-thee-well, and make all the changes your initial users want. When you think you have the whole thing just the way you want it, you find the one last bug that makes it virtually unusable. You go back to all your beta test sites and patch your software so that now you're sure it works correctly. After you've done all this you take the thing to market.

You sell three copies.

ENTER THE KID

What you need is better marketing. Marketing software is harder than writing it. It's a young man's game. You know this because of all the fresh young faces you've seen in the marketing departments of all the companies you've worked for.

So you get yourself a kid.

Our kid is clear-eyed, enthusiastic, and fresh out of business school. He's trained and ready to step in as marketing director and sell thousands of copies of our software. Actually, we were hoping to sell about ten copies this month, but, since we're paying this kid, we might as well listen to him.

THE KID LIKES VOLUME

The kid says the name of the game is volume.

"You have some good products here," the kid says. "You send people a tape, it takes ten minutes to install, and it works. Most commercial software won't do that. You ought to get this stuff out to as many people as you can."

THE KID GOES FOR IT

"What we're going to do is make it much easier for people to try the **FILE-SIZING TOOL**," the kid said.

"How are we going to do that?" we asked.

"Easy," says the kid. "We'll give it away."

"Great," we said. "Let's make up in volume what we lose on each individual sale."

"Listen. What we're going to do is let people try the **FILE-SIZING TOOL** for free. We'll give it to them for thirty days and if they like it, they'll buy it. Eighty percent of the people who try it now wind up buying it."

"Eighty percent, huh? How do you know that?"

"Market research. It's something I learned in school."

"Oh."

GENERATION RESEARCH GOES FOR IT

Fill in the coupon, sign and return the license agreement that we send you, and we'll send you a **FILE-SIZING TOOL** you can use free for thirty days. This offer is good only on the **FILE-SIZING TOOL** and only through September of 1983. You can look at our other products for thirty days too but they'll cost you our usual \$25 deposit. Of course, the deposit is refundable if you return the tape.

THE GENERATION RESEARCH LINE

We sell three products: The **FILE-SIZING TOOL** resizes all files on your Pick machine quickly and accurately. The **FILE-SIZING TOOL** costs \$475. **ELECTRONIC MAIL** gives users the ability to send, receive, and print letters. Try **ELECTRONIC MAIL**, and take control of corporate communications costs. **ELECTRONIC MAIL** costs \$999.00. **PROGRAMMING TOOLS** saves programmers valuable time in system development by providing advanced diagnostics for writing BASIC programs. **PROGRAMMING TOOLS** costs \$799.

All Generation Research software products run on all ADDS MENTOR, HONEYWELL ULTIMATE, and MICRODATA computers.

OK kid, take your shot.
Send me a tape.

Company Name: _____

Address: _____

City: _____ State: _____ Zip: _____

Attention: _____

Phone: (____) _____

Machine (check one) Software (check one)
☐ Adds ☐ File-Sizing Tool
☐ Honeywell ☐ Electronic Mail
☐ Microdata ☐ Programming Tools

Serial number if Microdata _____

Tape drive (check one)

☐ 800 BPI
☐ 1600 BPI

queries

The following query was originally published with no available answer in *Pragma* #2, page 31.

7. For reasons our customer engineer cannot explain, he has been taught to use a verb he calls CHOP before running the various ATP diagnostics. He manually creates the verb with the editor, placing a P in attribute one, and 2118 in attribute two. Also, he has been taught to enter six zeroes with the debugger at 280.135;3 when running the BLAST diagnostic under the 4.1 release. What does CHOP do, and why is the BLAST patch necessary?

The CHOP verb is used to reduce the overflow table in the case of a large disk. In a way it is self-defeating because using CHOP means the disk will not be tested at full capacity. The entry patched for BLAST is a Branch and Stack Location subroutine call to SLEEP, which would cause BLAST to "hang". By zeroing the branch instruction, BLAST never tries to SLEEP and will execute normally. [Answer submitted by Ken Baker, ADP Network Services, Dallas, TX]

Of the 24 Queries that have been featured in issues #1 through #4, seven now remain unanswered.

25. What is the function of the TCL command EED? I inadvertently entered EED instead of ED when modifying a BASIC program and ended up with all sorts of control characters in my code after it was filed. To avoid this problem I have deleted EED from the Master Dictionary. [Submitted by Brian C. Stone, CALPAC Inc., Los Angeles, CA]

The EED (or EEDIT) verb works just like the normal editor, except that it "compresses" items as they are filed in order to save disk space. Compression is accomplished by replacing any sequence of three or more blanks or asterisks in the text with a shorter string that includes a control character and a count. Once you have filed an item with EED, you should use that verb again for future edits, since the normal ED verb doesn't recognize compression codes and just displays them as control characters, making the item appear garbaged. The EBASIC verb can be used to compile compressed items, but the normal BASIC verb will treat all of those embedded control characters as syntax errors. The COPY verb will copy compressed items without complaining and leave them compressed. Use ECOPY to uncompress items. Note that these verbs are all briefly described in the Programmer's Reference Manual. A test that did an EED on a file of 274 different programs compressed each item an average of 15%, but don't assume an equal amount of disk space then became available, because the disk is allocated in frames of 512 bytes, and an item that is 800 bytes long and occupies two frames, for example, can still tie up two frames even when compressed to, say, 600 bytes.

□

• THE SIDEKICK •

A Microcomputer that talks to your Microdata

Now a desktop microcomputer
can communicate with your Microdata REALITY
via the standard PRISM port.

USE THE FILE TRANSFER MODE TO:

- PASS INFORMATION FROM MICRODATA FILES TO A FULL-FLEDGED WORK PROCESSOR SUCH AS WORDSTAR.
- SUPPORT REMOTE DATA ENTRY WITH FULL DATA VALIDATING AND SCREEN FORMATTING LOGIC. DIAL UP THE HOST AND TRANSMIT FILES PERIODICALLY.
- PERFORM A VARIETY OF TELECOMMUNICATIONS FUNCTIONS BY TRANSFERRING FILES TO/FROM THE MICRODATA AND USING THE PROGRAMMABLE PORTS ON THE MICROCOMPUTER.

USE THE INTERACTIVE MODE TO:

- LIGHTEN THE LOAD ON YOUR MICRODATA BY DISTRIBUTING PROCESSING WHILE STILL PERMITTING INQUIRY AND UPDATING OF MICRODATA FILES.
- PROVIDE MULTIPLE FUNCTIONS ON ONE PRISM® PORT. ONE MICROCOMPUTER CAN ACT AS A PRISM®, LOG TRANSACTIONS TO A FLOPPY DISK, AND DRIVE A PRINTER.

Programs are presently available for Televideo and Intertec Superbrain®. Others will be developed on request.

TURNKEY HARDWARE/SOFTWARE PACKAGES AVAILABLE.

Ben Ansbacher
AnsData Associates
P.O. Box 2598

Burlington, N.C. 27216-2598

800-334-6823 or 919-227-0123

PRISM and REALITY are trademarks of Microdata Corp.
WordStar is a trademark of MicroPro International Corp.
Superbrain is a trademark of Intertec Data Systems Corp.

FIXED ASSET/PREPAID EXPENSE ACCOUNTING MODULE

By Diogenes Computer Systems, Inc.

Specialists — Properties Management

Real Estate Construction

- * Easy to use
- * Hardcopy Validation/Backup
- * Flexible Accounting Periods
- * Book, Tax and State Calculations
- * Supports all IRS Approved Methods
- * 2 Years of Proven Reliability
- * Built in Security
- * Complete User and Program Documentation
- * Stand-Alone or Integrate with your G/L
- * Includes 1981 Economic Recovery Tax Act
- * Recapture Calculated Automatically
- * Depreciation Forecasting Module
- * Demonstrations Available in New York Metropolitan Area

AFFORDABLY PRICED AT \$2500.00

Integration and Implementation Available

For More Information —

Call — (201) 764-7282 8 A.M.—8 P.M.

Write — Diogenes Computer Systems, Inc.

P.O. Box 667

Highland Lakes, N.J. 07422

Software Licensing Inquiries Invited

SYSMAP

Part 5: Automation

This final article in a series on the use of cross-references discusses approaches to making SYSMAP a fully automated system.

SYSMAP is like a blank ledger page that requires the attention of a knowledgeable accountant to insure all of the rows and columns will be filled in with accurate, meaningful data. When a completely new software system is being written from scratch, SYSMAP files can be kept up to date as each piece of software is developed, although it generally must be an experienced programmer who scans each proc and program to decide what uses what. If a huge, undocumented set of software already exists, the tedious job of scanning all code to decide what to input to SYSMAP sounds particularly uninviting. What can be done so that SYSMAP files are automatically maintained?

One approach is to write programs that scan other programs to find references and automatically record them in the appropriate SYSMAP file. For example, a program could be written to scan procs and generate an item for SYSMAP's XP file anytime it finds a line in parentheses of the form (PROCLIB CALL), since one proc is obviously invoking another. There would have to be programs to scan procs, other programs and dictionary words, although the scanners wouldn't have to be 100% perfect, since their output could always be reviewed and manually adjusted. After all, it's fairly easy to write a program that recognizes READV X FROM I, 1 reads attribute one, but what about READV X FROM I, Y?

Using programs to scan other programs is better than a purely manual approach, but the burden of providing such programs is still on the user. A better solution is to design the compiler and other translators to automatically provide the cross-reference files. This puts the burden on the vendor, but most system translator programs are already designed to scan other programs. For example, the compiler generates a type of cross-reference now. If it and the other languages generated slightly more comprehensive data in the format of SYSMAP files, *it would be impossible to write software that wasn't automatically and completely cross-referenced by the system!* A system that automatically records the interactions of software also gains the ability to perform other spectacular feats to free the programmer from having to maintain the system. For example, imagine being able to indicate by a simple edit procedure that the dictionary word BALANCE for attribute five of a file should be deleted, along with deleting the attribute from all items in the file. Imagine having the system: (1) automatically recognize from its own cross-references that all attributes from six on must therefore be renumbered and that all programs that refer to those attributes must be adjusted and recompiled, and (2) then does it! That would be a true data base management system.

P

the computer room

Tape Types

The different ways in which files can be stored on magnetic tapes are described.

What are the different types of tapes? In other words, what are the different ways in which files can be saved on a tape? The primary methods for Pick machines are:

□ **FILE-SAVE tapes.** These tapes are generated by the FILE-SAVE proc, which saves on a tape a copy of every file on the disk. Since the complete contents of the disk is saved, making a FILE-SAVE tape is a good way to make a backup copy of all files, in case something should happen to the disk or to the computer causing data to be lost. A system with damaged or missing data can read back all files from a FILE-SAVE tape, thereby restoring the system and data to the state they were in at the time the tape was created. Even individual accounts, files or file items can be selectively restored from a FILE-SAVE tape. Also, the tapes can generally be loaded on other, similar models of the computer that wrote the tape.

□ **BINARY-SAVE tapes.** Like FILE-SAVE tapes, these contain a copy of all data on the disk, but are generated in such a way that individual accounts, files or file items cannot be selectively restored, and only the machine that created the tape can load it. Making a BINARY-SAVE tape is intended to be a quicker method of backing up a complete disk. See *Query #1 [Pragma #1, August 1982, page 32]* for a more detailed comparison of the two methods.

□ **ACCOUNT-SAVE tapes.** These are generated with the ACCOUNT-SAVE proc and are structured very much like a FILE-SAVE tape, but only contain the files found in one chosen account on the system. Data can be selectively restored from these tapes just like a FILE-SAVE tape. If all of the data defined in one account, but only that account, needs to be backed up, this is the kind of tape to create.

□ **T-DUMP tapes.** Each file on a T-DUMP tape contains items copied from a file on disk with the T-DUMP verb. A T-LOAD command can read the tape and load the items back into any other file. The T-DUMP and T-LOAD verbs support WITH clauses, so very selective dumping and loading can be done. This kind of tape is frequently used for mailing individual files or programs between computers.

□ **Miscellaneous tapes.** Tapes can also be written by the spooler or by a BASIC program that uses a WRITET statement, but these are fairly esoteric applications that are rarely, if ever, used at most installations.

P

Z-MAIL™

CAUTION:

**Reading this
could cause you
to acquire
the competitive
edge in office
communications**

Z-Mail is the ultimate electronic mail package. It allows the user to transmit messages in an automated office network, to another office system down the hall or halfway around the world.

In today's automated office, Z-Mail provides the tool to obtain the competitive edge. Discover Z-Mail before your competition does.

Z-Mail offers the future in electronic mail communications today!

Write or call today for detailed information on Z-Mail.



4851 Keller Springs Rd., Suite 208
Dallas, Texas 75248
214/931-3267

IBM's Personal Computer Talks to Your Minicomputer Via

R/NET



You can connect an IBM PC equipped with R/NET to your Reality, Ultimate, Mentor, Evolution, or Information relational data base machine. R/NET executes your existing software without modification. In addition to supporting cursor control for your interactive programs, you can either print the current screen contents or engage the printer for printing a multi-page report. And under operator or program control, R/NET will capture data sent to the IBM PC on its own disk storage.

Now you can let VisiCalc do your financial modeling. Execute EasyWriter on your dedicated microcomputer for your wordprocessing requirements. Your PC is multi-lingual, i.e., BASIC, Cobol, Pascal, Forth, FORTRAN, C, and the 8086 Macro Assembler. Application packages like General Ledger, Accounts Receivable, Accounts Payable, Inventory Control, Job Costing, and Payroll are available today.

R/NET is distributed on its own diskette and comes complete with documentation manual and interface cable for \$200. A typical PC configuration retails for \$3175. Make your next terminal an IBM Personal Computer, the most respected name in the industry.

cosmos

Direct your orders to: Cosmos Inc.,

Box AH, 123 Ferntree Dr. West, Morton, WA 98356
(206) 496-5974 • (206) 226-9362 for 24 hour answering.

Think Relational

• VisiCalc TM of VisiCorp • EasyWriter TM of Information Unlimited Software Inc. • R/NET TM of Cosmos Inc. • IBM TM of International Business Machines • Ultimate TM of Ultimate Corp. • Mentor TM of Applied Digital Data Systems • Evolution TM of Evolution Corp. • Information TM of Prime • Reality TM of Microdata Corp.

WKRPIk

FREE!
(No Obligation)

On Line Review of Pick®

Systems, Parts, Peripherals For Sale

For 300 or 1200 Baud (Vadic or Bell)

Dial: 513-232-5801
513-232-5802

LOGON to: WKRPIK (No Password)

Menu-Driven Software Will Let You:

- 1 Review FOR SALE Ads**
- 2 Request Follow-Up**
- 3 List Your Equipment With Mike**
- 4 Add Your Name to Our Mail List**
- 5 Other**

Shebesta Cincinnati

513-232-5000

(Pick, is this better?)

letters

If you use a Pick system, Pragma wants to hear from you. Have you developed applications of interest to other users? Do you plan to acquire new hardware? What features would you like to see in Pragma? Are you active in any type of user group organization? Write Pragma today. All letters to the Editors are welcome, and as many as possible will be published in the Letters Department in every issue.

Praise and a Correction from Prime

Let me congratulate you on your outstanding issue #4, as well as the previous issues. The material you presented was, in our opinion, very accurate. One small point. Prime Information, having been world marketed since 1978, has a worldwide installed base of over 1,100 systems as of this date, not 400 as listed [*A Survey of Hardware that Supports Pick-Style Software, May 1983, page 29*]. Again, congratulations!

Harold S. Rhodes
Prime Computer Inc.
Natick MA

We rechecked the original survey questionnaire returned by Prime and found that only 400 machines were indeed claimed. Perhaps 400 is the domestic figure? Prime has now made it clear that 1,100 worldwide is accurate, but has declined to describe how domestic and foreign sites divide that figure.

— Editors

How Many Users Are There? And Can They Exponentiate?

In the *Survey Says* article in your issue #4 of May, 1983 [page 3] you note that almost all of your survey replies were received from Microdata users. I must confess that I find it difficult to accept your argument that this is due to the overwhelming balance being in favor of Microdata sites. While it is true that Microdata still claims a large number of Pick-based sites, it is difficult to ignore the fact that among Ultimate, ADDS, Prime, Evolution and Intertechnique alone, there must be well over 3,000 systems, worldwide.

I would instead suggest that the reason for the apparent lack of response is that your publication fails to adequately address the interest areas of this growing Pick community. For example, your Wish List section seems to be heavily Microdata oriented since it consistently describes facilities which already exist on "standard" Pick implementations. In addition, many of the technical articles which appear either ignore the fact that there is a better way of doing things using the other Pick systems, or present a solution using only Microdata syntax and features (such as PQN procs and Screenpro), thus eliminating their usefulness for other possible users.

I should also note that I find consistent technical inaccuracies in Pragma. For example, in your recent issue which compares features [*A Comparison of BASIC Implementations, May 1983, page 18*], you note that there is no exponentiation

operator on either the ADDS or the Microdata systems. This is not, in fact, the case. Although the precision of these operators is limited, they are available on both systems. This kind of carelessness would tend to make a user suspicious about the accuracy of such articles.

In closing, I should say that I feel that Pragma is a good idea, and that with some refining and a greater amount of attention to the rest of the Pick community, it could be a very effective forum for communication within this group.

Joel A. Nirenberg
KTS Business Systems Inc.
Toronto ON

Unfortunately, Pragma is not (yet!) reaching every user in the world, otherwise the demographic breakdown of our readership would probably not be overwhelmed by Microdata sites, as it is today. We wish there were reliable figures on exactly what users are out there.

Your observation that we usually publish code and other items developed on a Microdata is accurate, although we feel readers should be careful not to ignore a proc just because the first line is PQN, or a program if it happens to contain an INPUT USING, since we try to publish software that is at least interesting and instructive, even if it may not be universally machine executable. We see only two good solutions for publishing more non-Microdata material: acquiring other machines for our staff (perhaps an ADDS and a Prime?), or acquiring articles from users of other machines. Guess which solution we feel we can justify. (We liked the way you quoted "standard". Maybe if a detailed, published standard did exist, it would be easier to detect incompatibilities?) In the mean time, we'll be glad to publish any explanation describing "a better way of doing things" using other systems.

As for exponentiation, we think you are confusing operators with functions, where an "operator" is a symbol such as a plus sign between two operands, and a "function" is a word such as SQRT followed by an argument list in parentheses. It is true that ADDS and Microdata can perform exponentiation, but only with the PWR function. In fact, all four implementations that we compared supported PWR, so PWR was not described in the survey, just as ABS and many other functions were ignored as described on page 18 of that issue. On the other hand, only Cosmos and Prime support exponentiation operators (either two asterisks or a caret), so those operators (which are not functions) were described as "not available" for Microdata and ADDS. We actually tried those operators on a Microdata, but the compiler reported an error. If they work on an ADDS, then they are a feature not described in the Mentor documentation we have.

— Editors

An Improved HEADUP

Your article *Self-Documenting Reports* in issue #4, May 1983 [page 27] is most interesting and useful. However, I make the following observations: first, the HEADUP proc must be written in "old" PQ proc versus "new" PQN proc on Microdata or the blanks will remain as attribute marks in the PROCREAD. Second, the restriction that no value comparisons (double quoted values) and certain single quoted words can not be utilized severely limits the usefulness of the program. I have enclosed a modification of the HEADUP program which solves the problem by eliminating the quote marks from the HEADING string. While this compromises the documentation slightly, the greater utility far outweighs any imperfections. It should be noted that line two of my version must be modified if the program is renamed and the new name has a different character length.

I greatly enjoy your magazine and find it extremely useful. Unlike many of your survey respondents I read Pragma cover to cover and find the time well spent. Keep up the good work.

Brian Stone
Calpac Inc.
Los Angeles CA

We tested your new version, and it works great, but we prefer our more general method of skipping the program name in the proc buffer, so we developed this third version:

```
HEADUP
001 PROCREAD COMMAND ELSE STOP
002 BLANK.POS = INDEX(COMMAND, " ", 1)
003 IF BLANK.POS = 0 THEN STOP
004 COMMAND = COMMAND[BLANK.POS+1, LEN(COMMAND)-BLANK.POS]
005 CMD = COMMAND
006 LOOP
007   Q.POS = INDEX(CMD, "'", 1)
008   WHILE Q.POS DO
009     CMD = CMD[1, Q.POS-1]:CMD[Q.POS+1, LEN(CMD)-Q.POS]
010   REPEAT
011   LOOP
012   Q.POS = INDEX(CMD, "'", 1)
013   WHILE Q.POS DO
014     CMD = CMD[1, Q.POS-1]:CMD[Q.POS+1, LEN(CMD)-Q.POS]
015   REPEAT
016   COMMAND = COMMAND: ' HEADING "' : "PAGE 'P' - " : CMD : " - '
      TL "' : " '
017   CHAIN COMMAND
018   END
```

This incorporates your quote-stripping method, although we used LOOPS instead of GO TOS. It is interesting to note that in Cosmos or Prime BASIC, each five-line LOOP can be replaced with a single CONVERT statement.

— Editors

Suppressing Histograms and the Editor, Desirable Functions

In answer to Wish #47 [*ISTAT and HASH-TEST Graph Suppression, Pragma #2, November 1982, page 23*], the STAT verb will produce the desired statistics without going through the detailed histogram.

Whenever I have accidentally typed in an editor command that would produce a long listing [see *Wish #64, Aborting Editor Commands, Pragma #4, May 1983, page 18*] and I wish to suppress the output and stay in the editor, I hit the break

key, type P and a carriage return, and hit the line feed key to resume editing. When the editor has finished "listing" the data, a prompt will reappear. Now repeat the break-P-line feed sequence to resume normal editing. The P command in the debugger or TCL suppresses output to the terminal and acts like a toggle. In the debugger, the message "ON" or "OFF" will remind you which state you are in. This technique is especially useful when working with a modem at slow baud rates.

Finally, there are three functions that I have seen in other computers' versions of BASIC that I wish were in DATA/BASIC. These are SWAP(X,Y), MIN(X,Y) and MAX(X,Y). SWAP would exchange the value of X with Y. MIN and MAX would return the minimum and maximum of the two arguments.

Mark A. Johnson
C.F. Guyon Inc.
Harrison NJ

Actually, HASH-TEST and STAT output is not really identical, especially when trying a test modulo, which STAT does not support. Suppressing the HASH-TEST graph would still be a useful feature.

Note that your editor suppression technique works even if the command being suppressed happens to exit the editor, such as when a self-referencing prestore command for deleting all items in a file exits the editor after the last item is deleted. Happily, P is accepted by the debugger even when the user privileges are only SYSO, at least on a Microdata.

There are many implementations of programming languages, such as FORTRAN, PL/I and COBOL, that have function libraries far more extensive than that available in DATA/BASIC, almost allowing us to make your wish list of functions arbitrarily long. What's really needed is a user function definition capability (sometimes seen in other BASIC implementations) that allows the programmer to define custom functions written in BASIC, either imbedded in a program as a local unit, or defined as an external, global library member that can be called by other programs. Note that your SWAP has been done using the syntax A:=B in other language implementations, including some BASICs. Also, why limit MIN and MAX to just two arguments? And why not allow dynamic arrays to be arguments for MIN and MAX? Or for that matter, how about if dynamic arrays are allowed as operands with arithmetic operators...

— Editors

More on Avoiding Saved Lists

Your article *Avoiding Saved Lists* in issue #4, May 1983 [page 26], was of extreme interest. I have discovered a similar example in the Microdata proc manual (release 3.2, page 107). Your proc is excellent for compiling and cataloging the same program in one step, but on a much more practical application I have been unable to get successful results and have continued using saved lists.

I am attaching a sample proc from one of the hundreds I currently use. An example of [avoiding saved lists in] this type of application would be greatly appreciated by not only me, but many of your other subscribers.

I would like to take this opportunity to praise your uncompile

ing program in the Utilities section. I appreciate this section and hope to see it continued.

Richard Chess
General Piping Products Inc.
Houston TX

The technique described in our article really works best at feeding item names one at a time to TCL-II verbs such as EDIT or BASIC, not ENGLISH verbs such as LIST or SORT, since TCL-II verbs are designed for processing explicit item lists one item at a time, while ENGLISH verbs can scan a whole file of implicit item names. (See your Programmer's Reference Manual for a description of the differences between TCL-I, TCL-II and ENGLISH verbs.) However, we still see other ways to improve the example you sent us, which with comments removed essentially looks like:

```
PQN
HSELECT ORDERS WITH COMPLETE "Y"
STON
HSAVE-LIST OPEN-ORDERS
P
HGET-LIST OPEN-ORDERS
STON
HSELECT ORDERS WITH CODE "1" "3"<
HSAVE-LIST OPEN-1
P
HGET-LIST OPEN-1
STON
HRUN BP PROCESS
P
HGET-LIST OPEN-ORDERS
STON
HSELECT ORDERS WITH CODE "2" "4"<
HSAVE-LIST OPEN-2
P
HGET-LIST OPEN-2
STON
HRUN BP PROCESS
P
```

It appears you are selecting "open" orders and then processing those orders with an odd CODE separate from those with an even CODE, even though the exact same program handles both cases. If CODE is attribute five, for example, then

```
001 PQN
002 HSELECT ORDERS WITH COMPLETE "Y"
003 H BY ODD.EVEN.CODE
004 STON
005 HRUN BP PROCESS
006 P

      ODD.EVEN.CODE
001 A
002 0
003
004
005
006
007
008 F:C2:5:R
009 L
010 10
```

would do the same job, but in a much more simple way. The ODD.EVEN.CODE item returns zero for even CODEs or one for odd CODEs. If odd CODEs must be processed first, not just separate, then BY-DSND ODD.EVEN.CODE should be used instead.

— Editors

Bigger Systems Found

In the article *Survey Says* in the May issue [Pragma #4, page 3], this question was asked: "Aren't there any bigger systems out there?" The hardware configuration for our Prime 850 consists of the following:

- 4 MB main memory
- 16 KB cache memory
- (3) 300 MB disk drives
- (2) 600 LPM line printers
- (2) 120 CPS printers
- (2) 300 LPM line printers
- (86) CRTs
- (96) ports
- (2) 75 IPS 800/1600 BPI tape drives

Maybe this configuration will qualify for being among the largest Pick look-alikes. As you can see, we are more interested in Prime Information articles than any other articles. Please try to have more articles on Prime Information.

James E. Prestridge
NASCO Inc.
Springfield TN

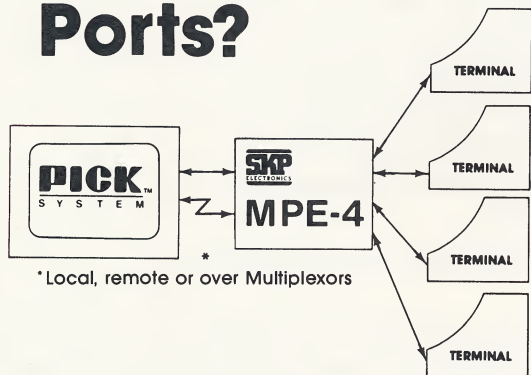
The 850 used to be the top of that line, but Prime has just informed us (by no less than an express mail service) that the 9950 is now available, offering "50% greater performance than the 850". Monisa Trice (of Drs. Groover, Christie and Merritt) and Ephraim Love (of JVC Company of America) also both wrote to describe their 2 MB Sequel systems. Perhaps one of you could run some benchmarks with NASCO to determine which system really appears biggest and fastest to the software?

— Editors

P

**HARDWARE
TO
BUY OR SELL?
GAIN EXPOSURE
TO A LARGE
RESPONSIVE
AUDIENCE
BY ADVERTISING
IN PRAGMA**

Need More Ports?



The MPE/4 allows 4 terminals to share one CPU Port. All electronic switching with optional log off. For more information call or write:



1232-E South Village Way
Santa Ana, CA 92705
(714) 972-1727

Pick Systems is a trademark of Pick Systems, Inc.

Microdata™ Sale

REALITY® 1600
128K Core
Reflex I
32 Ports
DP 300 LPM

REALITY® 6545
128K MOS
Reflex I
16 Ports
Printronix P-150

Other systems & equipment available.

The IJM Company
1438 Medford Avenue
Indianapolis, IN 46222

Jim Fisher
317/636-2677

Microdata™ Components FOR SALE

Marathon Disks • Core Memory
8-Ways • Tape • CRT • Printer

Call Jerry Buckley at 213-985-2680

Permuted Index to First Four Pragma Issues

The listing beginning on the next page indexes all articles, features and advertisements published in the first four issues of Pragma. To find an item of interest, first think of some word that describes what you are looking for, then find the word along the RIGHT side of the vertical DASHED line that runs down the middle of each half of the page. Once you have found a keyword, scan left on that line to find the pair of numbers indicating the issue and page where the article can be found. For example, if your keyword is ARIZONA, the index indicates you should check page 25 in issue #4. If your keyword is BILL OF MATERIAL, you should check issue #2, page 27.

The index has been generated by "permuting" a one line descriptor for each article so that all keywords in the descriptor appear sorted along a common line. For example, the article ZIP CODE FILE DESIGN would appear four times as

```

ZIP : CODE FILE DESIGN
ZIP CODE FILE : DESIGN
ZIP CODE : FILE DESIGN
          : ZIP CODE FILE DESIGN
    
```

Although this tiny sample is alphabetic along the vertical line, hundreds of other descriptors for other articles need to appear also, so the four entries shown here do not end up adjacent in the actual index.

The descriptor for each article is an arbitrarily long line consisting of fragments that identify:

Titles of articles (t)
Departments from the cover of Pragma (d)
Miscellaneous keywords (k)
Authors' last names (a)
Last names of people mentioned (n)
Vendors of advertisements (v)
Product names advertised (p)
Categories of advertised products (c)

The letters above in parentheses prefix each fragment inside every descriptor, and are summarized at the top of each index page. Many of these lower case prefixes are themselves alphabetized as keywords at the end of the index. Note that not every fragment in the list above appears in each descriptor.

Descriptors are truncated to fit into the column provided. To read the first words of a descriptor that is too long to fit in the column, look up the leftmost whole word in the current line. For example, the first descriptor for the keyword BENCHMARKS shows an article on page 21 in issue #1, but what is the title of the article? MEMORY is the leftmost whole word on that line, so looking up MEMORY in the index locates the same descriptor, but shifted more to the right within the column, revealing that the title begins with MORE MEMORY.

Ⓟ

4 18	EFAULT ATTRIBUTES, EDITOR	ABORT, CORRECTING TYPOS a	1 27	E, SCREENPRO MULTIVALUES,	BASIC INCLUDE, EDITOR LIN
1 27	S, STATEMENT LABELS, FILE	ACCESS, POWER-FAILS, LOGO	1 27	, ENGLISH PRINT LIMITING,	BASIC INPUT/OUTPUT, SYSTE
1 43	DK ENTERPRISES p) GAAP c)	ACCOUNTING PACKAGE	4 12	G, PART 1 d) UTILITIES k)	BASIC OBJECT CODE
2 4	DK ENTERPRISES p) GAAP c)	ACCOUNTING PACKAGE	2 38	t) SECURITY AND THE DATA/	BASIC PROGRAMMER d) ARTIC
3 7	SSET MANAGEMENT SYSTEM c)	ACCOUNTING PACKAGE	4 3	EY, CONFIGURATION SURVEY,	BASIC SURVEY, CONTEST n)
3 41	DK ENTERPRISES p) GAAP c)	ACCOUNTING PACKAGE	2 35	FROM IRVINE d) LETTERS k)	BAUD RATES a) DENNIS KEAT
4 8	DK ENTERPRISES p) GAAP c)	ACCOUNTING PACKAGE	1 27	LE SECURITY, SORT BREAKS,	BAUD RATES, PREVENTING LO
4 33	SSET MANAGEMENT SYSTEM c)	ACCOUNTING PACKAGE	3 31	OGRAM AND GARBAGECOLLECT,	BAUD SWITCHES, U508E
4 41	p) FIXED ASSET MODULE c)	ACCOUNTING PACKAGE	4 3	LD, JANICE BARTLETT, GAIL	BELLON, JAY BONHAM, ANNA
3 27	ENGLISH HEADINGS, SPOOLER	ACCOUNTING, LOCK LIMITS,	4 3	R HARPEL, RAY WONG, LOUIS	BELTJENS, BRUCE SLAWINSKI
3 44	ACCU/SOFT ENTERPRISES p)	ACCU-PLOT II c) GRAPHICS	1 21	RE MEMORY, FEWER READS d)	BENCHMARKS
4 44	ACCU/SOFT ENTERPRISES p)	ACCU-PLOT II c) GRAPHICS	3 25	MORE BASIC COMPARISONS d)	BENCHMARKS k) LOOP, LOCAT
3 36	t)	ACCU/PLOT UPGRADES d) LET	4 17	S, RIGHT JUSTIFICATION d)	BENCHMARKS k) SORTING
3 44	d) AD v)	ACCU/SOFT ENTERPRISES p)	2 20	t) LOOP VS. LOCATE d)	BENCHMARKS n) DAVIN LAWSO
4 44	d) AD v)	ACCU/SOFT ENTERPRISES p)	2 27	FACTURING SYSTEM, PART 2:	BILL OF MATERIAL INPUT d)
4 29	TOUGH, LINDA SPELMAN, LEE	ADAMSON, JEANINE CHRISTIA	1 32	d) QUERIES k)	BINARY AND FILE SAVE/REST
3 33	d) AD v)	ADDEPT p) CONSULTING SERV	3 40	M THAT LEARNS d) GAMES k)	BINARY TREES
3 17	R PROFILE k) MICRODATA TO	ADDS CONVERSION, DISTRIBU	2 10	t)	BLACK BOX FORMATTING d) A
4 18	PRIME VS. MICRODATA VS.	ADDS n) ROGER HARPEL, DAV	3 38	t) GENERATING	BLANK FORMS d) ARTICLE k)
2 31	BLE DISK SPACE, PRISM VS.	ADDS, WHERE LINKS, ITEM L	2 31	ACE, HOLD FILES, CHOP AND	BLAST, ERRMSG [13], SECUR
1 43	v) SEMAPHORE p) PRAGMA c)	ADVERTISING SPACE	1 27	SES, SORT-LABEL, ON ELSE,	BLOCK-CONVERT, FILE SECUR
2 8	v) SEMAPHORE p) PRAGMA c)	ADVERTISING SPACE	2 18	d) SUBSCRIBERS n) MIKE	BLUE, PAUL CRAVEN, ALAN T
3 30	v) SEMAPHORE p) PRAGMA c)	ADVERTISING SPACE	4 3	ARTLETT, GAIL BELLON, JAY	BONHAM, ANNA CARSEN, STEV
1 2	UBSCRIPTIONS, SUBMITTALS,	ADVERTISING, READER SERVI	1 27	d) WISH LIST k) SLEEP,	BREAK-ON, FRAME FAULTS, E
2 2	UBSCRIPTIONS, SUBMITTALS,	ADVERTISING, READER SERVI	2 31	GORITHM, RELEASE NUMBERS,	BREAK-ON, SYSPROG VS. SP,
3 4	UBSCRIPTIONS, SUBMITTALS,	ADVERTISING, READER SERVI	1 27	VERT, FILE SECURITY, SORT	BREAKS, BAUD RATES, PREVE
4 4	UBSCRIPTIONS, SUBMITTALS,	ADVERTISING, READER SERVI	4 29	DS, TERRY MULHERN, DENNIS	BROWN, ROGER HARPEL, PAUL
4 30	t) ANOTHER EDIT	AIDE d) LETTERS a) JACK H	1 36	2 SCREENPRO d) ARTICLE k)	BUG
3 6	t) EDIT	AIDES d) ARTICLE k) DEFAU	1 30	t)	BUILDING ENGLISH HEADINGS
3 39	AD v) SOFTWARE GROUP p)	AIDS c) APPLICATIONS GENE	3 33	d) AD v)	BURNS & ASSOC p) SAM c) P
4 7	AD v) SOFTWARE GROUP p)	AIDS c) APPLICATIONS GENE	4 41	d) AD v)	BURNS & ASSOC p) SAM c) P
2 31	S, ITEM LOCKS, CHARGEBACK	ALGORITHM, RELEASE NUMBER	3 33	DATA REALITY c) WANTED TO	BUY
1 32	RY AND FILE SAVE/RESTORE,	ALL, 4.1 SPOOLER	4 25	VALLEY, ARIZONA, NORTHERN	CALIFORNIA n) BOB CLEARFI
3 32	d) AD v)	ALPHATEST p) MICRODATA RE	2 22	, E REGISTER, CORRELATIVE	CALLS, ON-LINE DOCUMENTAT
2 45	N, MALIA CULBRETH, JOSEPH	AMERES	4 25	t) AN UNDOCUMENTED EDITOR	CAPABILITY d) ARTICLE k)
2 18	AVEN, ALAN THOMPSON, LYNN	ANDERSON	1 24	d) AD v) SEMAPHORE p)	CARGO c) SPREADSHEET
4 3	A, WESLEY WOODLAND, KAREN	ANDERSON	2 46	d) AD v) SEMAPHORE p)	CARGO c) SPREADSHEET
3 40	t)	ANIMAL, A PROGRAM THAT LE	3 47	d) AD v) SEMAPHORE p)	CARGO c) SPREADSHEET
4 41	d) AD v)	ANSDATA ASSOC p) THE SIDE	4 3	BELLON, JAY BONHAM, ANNA	CARSEN, STEVE MATTSOON, TO
3 35	t) SOME WISHES ARE	ANSWERED d) LETTERS k) SL	4 25	EARFIELD, JIM REILLY, JIM	CATES, SHELLY FECKER, JO
3 39	SOFTWARE GROUP p) AIDS c)	APPLICATIONS GENERATOR	2 31	WHERE LINKS, ITEM LOCKS,	CHARGEBACK ALGORITHM, REL
3 47	v) CHRONON DATA p) HAL c)	APPLICATIONS GENERATOR	2 22	PQ VS. PQN, PAGE NUMBERS,	CHECK SPOOLER, ISTAT AND
4 7	SOFTWARE GROUP p) AIDS c)	APPLICATIONS GENERATOR	3 32	t) SHARED SITE	CHECKLIST d) COMPUTER ROO
4 48	v) CHRONON DATA p) HAL c)	APPLICATIONS GENERATOR	1 26	t) HOW TO FLAG MISSING	CHECKS d) ARTICLE k) ENGL
3 6	ARTICLE k) DEFAULT PROC	ARGUMENTS a) MICHAEL ROSS	4 24	EHAN, KEVIN COOK, RICHARD	CHESS, RICHARD USRY, BRIA
4 25	ROUPS k) DELAWARE VALLEY,	ARIZONA, NORTHERN CALIFOR	2 31	k) WORKSPACE, HOLD FILES,	CHOP AND BLAST, ERRMSG [1
2 22	S, ON-LINE DOCUMENTATION,	ASSERTIONS, SCREENPRO COM	4 29	MAN, LEE ADAMSON, JEANINE	CHRISTIANO, DICK GOULD
3 7	AD v) MULTIDATA p) FIXED	ASSET MANAGEMENT SYSTEM c	3 47	d) AD v)	CHRONON DATA p) HAL c) AP
4 33	AD v) MULTIDATA p) FIXED	ASSET MANAGEMENT SYSTEM c	4 48	d) AD v)	CHRONON DATA p) HAL c) AP
4 41	COMPUTER SYSTEMS p) FIXED	ASSET MODULE c) ACCOUNTIN	4 25	ORTHERN CALIFORNIA n) BOB	CLEARFIELD, JIM REILLY, J
2 22	CONTIGUOUS OUTPUT, STRING	ASSIGNMENT, SCREENPRO MUL	4 12	UTILITIES k) BASIC OBJECT	CODE
3 26	GGED OUTPUT d) ARTICLE k)	ATTRIBUTE HEADINGS	1 4	t) ZIP	CODE FILE DESIGN d) ARTIC
3 27	FIER LENGTH, GROUP LOCKS,	ATTRIBUTE SECURITY, ENGLI	4 25	ILITY d) ARTICLE k) SEMI-	COLON OPERATOR n) ED SHEE
4 18	PY NULL SUPPRESS, DEFAULT	ATTRIBUTES, EDITOR ABORT,	3 28	t) GENERATING MONTHLY	COLUMN HEADINGS d) COMMAN
1 45	TERACTIVE SYSTEMS p) TCL-	AUDITOR, REPORT-GEN, SCRE	2 26	t) PROC CONVERSIONS d)	COMMAND FILES
2 5	d) AD v) AUROTECH p)	AUROPLAN c) SPREADSHEET	1 30	ISH HEADINGS WITH PROC d)	COMMAND FILES k) DEBUGGIN
3 2	d) AD v) AUROTECH p)	AUROPLAN c) SPREADSHEET	3 28	ONTHLY COLUMN HEADINGS d)	COMMAND FILES k) ENGLISH
4 47	d) AD v) AUROTECH p)	AUROPLAN c) SPREADSHEET	4 26	LISTS WITH PQ-RESELECT d)	COMMAND FILES k) SELECT
2 5	d) AD v)	AUROTECH p) AUROPLAN c) S	2 22	ON, ASSERTIONS, SCREENPRO	COMMAND, EDITOR B, PQ-COM
3 2	d) AD v)	AUROTECH p) AUROPLAN c) S	3 31	TO ENGLISH, PART 2: MORE	COMMANDS d) ARTICLE
4 47	d) AD v)	AUROTECH p) AUROPLAN c) S	3 27	UNTING, LOCK LIMITS, SLOW	COMMANDS, MF CONVERSIONS,
2 8	DELTA DATA CONSULTANTS p)	AUTOMATIC LINE OPERATOR c	1 48	d) AD v) DYNATRONICS p)	COMMUNICATIONS CONTROLLER
2 8	d) AD v)	AUTOMATIC PROGRAMMING p)	2 11	k) FILE CONSTRUCTION AND	COMPACTION
3 8	d) AD v)	AUTOMATIC PROGRAMMING p)	4 18	t) A	COMPARISON OF BASIC IMPLE
4 41	d) AD v)	AUTOMATIC PROGRAMMING p)	3 25	t) MORE BASIC	COMPARISONS d) BENCHMARKS
2 31	T, ERRMSG [13], SECURITY,	AVAILABLE DISK SPACE, PRI	2 22	RO COMMAND, EDITOR B, PQ-	COMPILE STON, PAGE, UNDOC
4 3	ITORIAL k) READER SURVEY,	AVAILABLE HARDWARE SURVEY	2 41	v) INTERACTIVE SYSTEMS p)	COMPU-SHEET c) SPREADSHEE
4 26	t)	AVOIDING SAVED LISTS WITH	3 1	v) INTERACTIVE SYSTEMS p)	COMPU-SHEET c) SPREADSHEE
2 22	SCREENPRO COMMAND, EDITOR	B, PQ-COMPILE STON, PAGE,	4 31	v) INTERACTIVE SYSTEMS p)	COMPU-SHEET c) SPREADSHEE
2 22	LER, ISTAT AND HASH-TEST,	BACKSPACE	3 8	d) AD v)	COMPUTER DISTRIBUTORS c)
4 14	t) PACIFIC VALLEY	BANK d) USER PROFILE k) M	1 37	d) AD v)	COMPUTER DISTRIBUTORS p)
4 3	BSOON, TJ LEIFIELD, JANICE	BARTLETT, GAIL BELLON, JA	2 32	t) THE TROUBLE TREE d)	COMPUTER ROOM k) DIAGNOST
3 28	MAND FILES k) ENGLISH VS.	BASIC	3 32	SHARED SITE CHECKLIST d)	COMPUTER ROOM k) DISASTER
3 25	t) MORE	BASIC COMPARISONS d) BENC	4 29	t) PRINTER TRADE-OFFS d)	COMPUTER ROOM k) HARDWARE
4 18	t) A COMPARISON OF	BASIC IMPLEMENTATIONS d)	1 34	d)	COMPUTER ROOM k) TAPE MOU

1 37	BUTORS p)	IBM SERIES/1 c)	COMPUTER SYSTEM	1 33	H WELL d)	ARTICLE k)	FILE	DESIGN
1 42	R SYSTEMS p)	EVOLUTION c)	COMPUTER SYSTEM	1 4		t)	ZIP CODE FILE	DESIGN d) ARTICLE k) DATA
1 47	) MICRODATA p)	REALITY c)	COMPUTER SYSTEM	2 32	TREE d)	COMPUTER ROOM k)		DIAGNOSTIC DOCUMENTATION
1 42	d)	AD v)	EVOLUTION	4 6		t)	SYSMAP, PART 4:	DICTS AND PROCS d) ARTICL
4 41	d)	AD v)	DIODENES	4 41		d)	AD v)	DIODENES COMPUTER SYSTEMS
2 8	d)	AD v)	IRVINE	3 3	L k)	READER SURVEY, IDBMA		DIRECTORY, PRODUCT EVALUA
3 33	d)	AD v)	IRVINE	3 32	KLIST d)	COMPUTER ROOM k)		DISASTER RECOVERY
4 3		t)	COMPUTER-GENERATED INDEX	2 10	FORMATTING d)	ARTICLE k)		DISK CONTROLLER
1 29		t)	COMPUTING MODULOS WITH EN	2 31	[13], SECURITY, AVAILABLE			DISK SPACE, PRISM VS. ADD
3 36		t)	COMPUTING MODULOS d) LETT	1 34	ER ROOM k)	TAPE MOUNTING/		DISMOUNTING
2 35		t)	COMPUTING MODULOS d) LETT	2 22	) WISH LIST k)	PASSWORDS,		DISPLAY STATEMENT, SCREEN
4 29	COMPUTER ROOM k)	HARDWARE	CONFIGURATION	3 17	ODATA TO ADDS CONVERSION,			DISTRIBUTOR, WIZARD n) ST
3 3	TORY, PRODUCT EVALUATION,		CONFIGURATION SURVEY	3 8	d)	AD v)	COMPUTER	DISTRIBUTORS c) PERSONNEL
4 3	AVAILABLE HARDWARE SURVEY,		CONFIGURATION SURVEY, BAS	1 37	d)	AD v)	COMPUTER	DISTRIBUTORS p) IBM SERIE
2 11	FILE d)	UTILITIES k)	FILE	1 22	TEM d)	ARTICLE k)	PROGRAM	DOCUMENTATION
2 8	d)	AD v)	DELTA DATA	2 32	MPUTER ROOM k)	DIAGNOSTIC		DOCUMENTATION
1 41	d)	AD v)	SEMAPHORE p)	3 5	LITY TECHNICAL JOURNAL c)			DOCUMENTATION
2 48	d)	AD v)	SEMAPHORE p)	4 5	LITY TECHNICAL JOURNAL c)			DOCUMENTATION
3 33	d)	AD v)	ADDEPT p)	4 35	S p)	PICK POCKET GUIDE c)		DOCUMENTATION
3 48	d)	AD v)	SEMAPHORE p)	2 22	ORRELATIVE CALLS, ON-LINE			DOCUMENTATION, ASSERTIONS
3 33	d)	AD v)	PICK SYSTEMS c)	4 27		t)	SELF-	DOCUMENTING REPORTS d) AR
4 3	ION SURVEY, BASIC SURVEY,		CONTEST n) ROGER HARPEL,	2 40	VEN DAVIES-MORRIS, THOMAS			DOWLING, DON KATZMAN, PEG
2 22	DE, FILE ENGLISH OUTPUT,		CONTIGUOUS OUTPUT, STRING	1 48		d)	AD v)	DYNATRONICS p) COMMUNICAT
2 10	ATTING d)	ARTICLE k)	DISK	3 31	LECT, BAUD SWITCHES, U508			E
1 48	TRONICS p)	COMMUNICATIONS	CONTROLLER c)	2 22	MULTIVALUES, PAGE EDITOR,			E REGISTER, CORRELATIVE C
4 14	ILE k)	MICRODATA TO PRIME	CONVERSION n) ALEXANDER L	3 8		d)	AD v)	EASTCOMP II c) USED HARDW
2 16	FILE k)	REALITY TO SEQUEL	CONVERSION, CPA n) JOE KE	4 34		d)	AD v)	EASTCOMP II c) USED HARDW
3 17	FILE k)	MICRODATA TO ADDS	CONVERSION, DISTRIBUTOR,	4 30		t)	ANOTHER	EDIT AIDE d) LETTERS a) J
2 26		t)	PROC	3 6		t)		EDIT AIDES d) ARTICLE k)
1 29	ITH ENGLISH d)	ARTICLE k)	CONVERSIONS, CORRELATIVES	4 18	RESS, DEFAULT ATTRIBUTES,			EDITOR ABORT, CORRECTING
2 22	PLAY STATEMENT, SCREENPRO		CONVERSIONS, EDITOR DE, F	2 22	TIONS, SCREENPRO COMMAND,			EDITOR B, PQ-COMPILE STON
3 27	LIMITS, SLOW COMMANDS, MF		CONVERSIONS, SELECT ONE,	4 25		t)	AN UNDOCUMENTED	EDITOR CAPABILITY d) ARTI
1 27	RT-LABEL, ON ELSE, BLOCK-		CONVERT, FILE SECURITY, S	2 22	T, SCREENPRO CONVERSIONS,			EDITOR DE, FILE ENGLISH O
3 34		t)	CONVERTING PAINT TO PROGR	3 27	T k)	HEADING EXPRESSIONS,		EDITOR DE, IDENTIFIER LEN
4 24	BERS n)	ED SHEEHAN, KEVIN	COOK, RICHARD CHESS, RICH	3 27	CT ONE, INPUT TIME LIMIT,			EDITOR F
2 39		t)	COOKIE d) GAMES k) FORTUN	1 27	LTIVALUES, BASIC INCLUDE,			EDITOR LINE SPLITTING, LO
4 18		d)	WISH LIST k)	1 27	ON, FRAME FAULTS, ERRMSG,			EDITOR PRESTORE, SCREENPR
3 8	ER, JOEL NIRENBERG, BRUCE		CORBETT, FRANK WILEY	3 35	ERED d)	LETTERS k)	SLEEP,	EDITOR PRESTORE, SPLITIN
4 18	ATTRIBUTES, EDITOR ABORT,		CORRECTING TYPOS a) MARK	2 22	UNENTED FEATURES, LOCATE,			EDITOR U, PQ VS. PGN, PAG
2 22	PAGE EDITOR, E REGISTER,		CORRELATIVE CALLS, ON-LIN	2 22	REENPRO MULTIVALUES, PAGE			EDITOR, E REGISTER, CORRE
1 26	CKS d)	ARTICLE k)	ENGLISH	1 3	t)	WELCOME TO PRAGMA d)		EDITORIAL k) PRAGMATISM n
1 29	)	ARTICLE k)	CONVERSIONS,	2 3		t)	WE HAVE LIFTOFF d)	EDITORIAL k) READER PREFE
3 38	RMS d)	ARTICLE k)	ENGLISH	4 3		t)	SURVEY SAYS... d)	EDITORIAL k) READER SURVE
4 18	EMENTATIONS d)	ARTICLE k)	CORRELATIVES a) HENRY MUD	3 3	ARE MEDIUM, WELL DONE? d)			EDITORIAL k) READER SURVE
1 46		d)	AD v)	4 24	HESS, RICHARD USRY, BRIAN			EHLERS, W SHAVE
3 37		d)	AD v)	1 27	RELEASES, SORT-LABEL, ON			ELSE, BLOCK-CONVERT, FILE
4 43		d)	AD v)	1 15	RTICLE k)	MRP, SCREENPRO,		ENGINEERING
4 2		d)	AD v)	2 27	ERIAL INPUT d)	ARTICLE k)		ENGINEERING
1 14	t)	DERIVING YEAR-TO-DATE	COSMOS VS. PRIME VS. MICR	1 26	SING CHECKS d)	ARTICLE k)		ENGLISH CORRELATIVES
4 3	MATTSON, TOM WELCH, PETER		COSMOS p) R/NET c) NETWOR	3 38	BLANK FORMS d)	ARTICLE k)		ENGLISH CORRELATIVES a) H
2 16	ITY TO SEQUEL CONVERSION,		COSMOS p) R/NET c) NETWOR	4 27	ING REPORTS d)	ARTICLE k)		ENGLISH HEADINGS
2 18	RIBERS n)	MIKE BLUE, PAUL	COSMOS p) R/NET c) NETWOR	1 30		t)	BUILDING	ENGLISH HEADINGS WITH PRO
2 31	BREAK-ON, SYSPROG VS. SP,		COSMOS p) REVELATION c) O	3 27	OCKS, ATTRIBUTE SECURITY,			ENGLISH HEADINGS, SPOOLER
1 22	t)	SYSMAP, PART 1: A	COUNTS d) ARTICLE k) FILE	2 22	VERSIONS, EDITOR DE, FILE			ENGLISH OUTPUT, CONTIGUOU
3 12	LIST.POINTERS: A PROC FOR		COY, CHERYL GUMINIK, SHIR	1 27	SPLITTING, LOGON LIMITS,			ENGLISH PRINT LIMITING, B
2 45	HUBER, MARK NELSON, MALIA		CPA n) JOE KEMPTER	3 28	DINGS d)	COMMAND FILES k)		ENGLISH VS. BASIC
4 29	BROWN, ROGER HARPEL, PAUL		CRAVEN, ALAN THOMPSON, LY	1 29	t)	COMPUTING MODULOS WITH		ENGLISH d) ARTICLE k) CON
2 8	d)	AD v)	CREATING FILES	2 24		t)	AN INTRODUCTION TO	ENGLISH, PART 1: JARGON d
1 4	FILE DESIGN d)	ARTICLE k)	CROSS-REFERENCE SYSTEM d)	3 31		t)	AN INTRODUCTION TO	ENGLISH, PART 2: MORE COM
3 47	d)	AD v)	CHRONON	4 30		t)	AN INTRODUCTION TO	ENGLISH, PART 3: FINDING
4 48	d)	AD v)	CHRONON	1 4	DESIGN d)	ARTICLE k)	DATA	ENTRY
2 38	t)	SECURITY AND THE	DATA p) HAL c) APPLICATIO	4 32	M, PART 4: PURCHASE ORDER			ENTRY d) ARTICLE
1 14	t)	DERIVING YEAR-TO-	DATA p) HAL c) APPLICATIO	2 31	LD FILES, CHOP AND BLAST,			ERRMSG [13], SECURITY, AV
2 40	d)	SUBSCRIBERS n) STEVEN	DATE COUNTS d) ARTICLE k)	1 27	, BREAK-ON, FRAME FAULTS,			ERRMSG, EDITOR PRESTORE,
2 22	ENPRO CONVERSIONS, EDITOR		DAVIES-MORRIS, THOMAS DOW	3 3	IDBMA DIRECTORY, PRODUCT			EVALUATION, CONFIGURATION
3 27	ADING EXPRESSIONS, EDITOR		DE, FILE ENGLISH OUTPUT,	1 42		d)	AD v)	EVOLUTION COMPUTER SYSTEM
1 30	PROC d)	COMMAND FILES k)	DEBUGGING	1 42	UTION COMPUTER SYSTEMS p)			EVOLUTION c) COMPUTER SYS
4 18	ST k)	COPY NULL SUPPRESS,	DEFAULT ATTRIBUTES, EDITO	1 36		t)	PATCHING THE	EXIT PROBLEM IN 3.2 SCREE
3 6	EDIT AIDES d)	ARTICLE k)	DEFAULT PROC ARGUMENTS a)	3 31	d)	QUERIES k)	LISTU, PROC	EXITS, PROGRAM AND GARBAG
4 25	d)	LOCAL USER GROUPS k)	DELAWARE VALLEY, ARIZONA,	3 27	d)	WISH LIST k)	HEADING	EXPRESSIONS, EDITOR DE, I
2 11		t)	TRIM.	3 27	INPUT TIME LIMIT, EDITOR			F
2 8		d)	AD v)	1 27	BELS, FILE ACCESS, POWER-			FAILS, LOGON HISTORY
1 14		t)	DERIVING YEAR-TO-DATE COU	1 27	k)	SLEEP, BREAK-ON, FRAME		FAULTS, ERRMSG, EDITOR PR
4 42	t)	SWAT! d)	GAMES n) RENE	1 21		t)	MORE MEMORY,	FEWER READS d) BENCHMARKS
1 14	COUNTS d)	ARTICLE k)	FILE	3 25	S k)	LOOP, LOCATE, OCONV,		FIELD n) STEVE KOWARSKY,
			DESIGN					

1 27 LOGONS, STATEMENT LABELS,	FILE ACCESS, POWER-FAILS,	3 33 NE COMPUTER c) PERIPHERAL	HARDWARE
2 11 D PROFILE d) UTILITIES k)	FILE CONSTRUCTION AND COM	3 37 ZENTEC p) REFLEX c) USED	HARDWARE
1 14 DATE COUNTS d) ARTICLE k)	FILE DESIGN	4 11 d) AD v) SHEBESTA c) USED	HARDWARE
1 33 S HASH WELL d) ARTICLE k)	FILE DESIGN	4 34 MICRODATA REALITY c) USED	HARDWARE
1 4 t) ZIP CODE	FILE DESIGN d) ARTICLE k)	4 34 ZENTEC p) REFLEX c) USED	HARDWARE
2 22 O CONVERSIONS, EDITOR DE,	FILE ENGLISH OUTPUT, CONT	4 34 AD v) EASTCOMP II c) USED	HARDWARE
2 14 t) SYSMAP, PART 2:	FILE FORMAT INPUT d) ARTI	4 29 -OFFS d) COMPUTER ROOM k)	HARDWARE CONFIGURATION
3 15 SYSMAP, PART 3: REMAINING	FILE INPUT d) ARTICLE	4 3 READER SURVEY, AVAILABLE	HARDWARE SURVEY, CONFIGUR
1 32 d) QUERIES k) BINARY AND	FILE SAVE/RESTORE, ALL, 4	4 29 t) A SURVEY OF	HARDWARE THAT SUPPORTS PI
1 27 , ON ELSE, BLOCK-CONVERT,	FILE SECURITY, SORT BREAK	4 41) THE SIDEKICK c) NETWORK	HARDWARE/SOFTWARE
1 15 SYSTEM, PART 1: THE PARTS	FILE d) ARTICLE k) MRP, S	3 25 n) STEVE KOMARSKY, ROGER	HARPEL
2 26 OC CONVERSIONS d) COMMAND	FILES	4 18 CRODATA VS. ADDS n) ROGER	HARPEL, DAVE OSTBY, ALLIS
2 31 SYSPROG VS. SP, CREATING	FILES	4 29 HERN, DENNIS BROWN, ROGER	HARPEL, PAUL DAHILL, SHIR
4 30 ENGLISH, PART 3: FINDING	FILES d) ARTICLE	4 3 SURVEY, CONTEST n) ROGER	HARPEL, RAY WONG, LOUIS B
1 30 INGS WITH PROC d) COMMAND	FILES k) DEBUGGING	4 18 Y, ALLISON HARTMANN, DAVE	HARRAR, CAROL TOMLINSON,
3 28 OLUMN HEADINGS d) COMMAND	FILES k) ENGLISH VS. BASI	4 18 RPEL, DAVE OSTBY, ALLISON	HARTMANN, DAVE HARRAR, CA
4 26 TH PQ-RESELECT d) COMMAND	FILES k) SELECT	4 3 , CHERYL GUMINIK, SHIRLEY	HARVEY, PHIL HUNTINGDON,
2 31 UERIES k) WORKSPACE, HOLD	FILES, CHOP AND BLAST, ER	1 33) MAKING ITEM IDENTIFIERS	HASH WELL d) ARTICLE k) F
4 30 CTION TO ENGLISH, PART 3:	FINDING FILES d) ARTICLE	2 22 CHECK SPOOLER, ISTAT AND	HASH-TEST, BACKSPACE
3 7 d) AD v) MULTIDATA p)	FIXED ASSET MANAGEMENT SY	3 27 d) WISH LIST k)	HEADING EXPRESSIONS, EDIT
4 33 d) AD v) MULTIDATA p)	FIXED ASSET MANAGEMENT SY	3 26 T d) ARTICLE k) ATTRIBUTE	HEADINGS
4 41 GENES COMPUTER SYSTEMS p)	FIXED ASSET MODULE c) ACC	4 27 RTS d) ARTICLE k) ENGLISH	HEADINGS
1 26 t) HOW TO	FLAG MISSING CHECKS d) AR	1 30 t) BUILDING ENGLISH	HEADINGS WITH PROC d) COM
2 43 LEMOT, SERGE KRISIUK, LEO	FOHL	3 28 GENERATING MONTHLY COLUMN	HEADINGS d) COMMAND FILES
2 14 t) SYSMAP, PART 2: FILE	FORMAT INPUT d) ARTICLE	3 27 TRIBUTE SECURITY, ENGLISH	HEADINGS, SPOOLER ACCOUNT
2 10 t) BLACK BOX	FORMATTING d) ARTICLE k)	4 25 PICK, MIKE SIBLEY, HENRY	HERMAN, LISA LEVSEN
3 38 t) GENERATING BLANK	FORMS d) ARTICLE k) ENGLI	4 25 TES, SHELLEY FRECKER, JODI	HILGENBERG, MARK PICK, MI
2 39 t) COOKIE d) GAMES k)	FORTUNE TELLING	4 30 D MARKS, BRIAN HILL, MARK	HILL
1 27 LIST k) SLEEP, BREAK-ON,	FRAME FAULTS, ERRMSG, EDI	4 30 ETTI, HOWARD MARKS, BRIAN	HILL, MARK HILL
4 25 REILLY, JIM CATES, SHELLEY	FRECKER, JODI HILGENBERG,	1 27 CCESS, POWER-FAILS, LOGON	HISTORY
1 43 AD v) KDK ENTERPRISES p)	GAAP c) ACCOUNTING PACKAG	2 31 d) QUERIES k) WORKSPACE,	HOLD FILES, CHOP AND BLAS
2 4 AD v) KDK ENTERPRISES p)	GAAP c) ACCOUNTING PACKAG	2 45 d) SUBSCRIBERS n) ARTHUR	HUBER, MARK NELSON, MALIA
3 41 AD v) KDK ENTERPRISES p)	GAAP c) ACCOUNTING PACKAG	4 3 NIK, SHIRLEY HARVEY, PHIL	HUNTINGDON, JOSEPH SIROTA
4 8 AD v) KDK ENTERPRISES p)	GAAP c) ACCOUNTING PACKAG	1 37 COMPUTER DISTRIBUTORS p)	IBM SERIES/1 c) COMPUTER
2 16 t)	GALLINSKI HAMBURG d) USER	3 3 ITORIAL k) READER SURVEY,	IDBMA DIRECTORY, PRODUCT
1 38 HE SHELL GAME d) GAMES k)	GAMBLING	3 27 G EXPRESSIONS, EDITOR DE,	IDENTIFIER LENGTH, GROUP
1 38 t) THE SHELL	GAME d) GAMES k) GAMBLING	1 33 t) MAKING ITEM	IDENTIFIERS HASH WELL d)
3 40 A PROGRAM THAT LEARNS d)	GAMES k) BINARY TREES	4 18 t) A COMPARISON OF BASIC	IMPLEMENTATIONS d) ARTICL
2 39 t) COOKIE d)	GAMES k) FORTUNE TELLING	4 27 d) QUERIES k)	IMPROVING THROUGHPUT
1 38 t) THE SHELL GAME d)	GAMES k) GAMBLING	1 27 EENPRO MULTIVALUES, BASIC	INCLUDE, EDITOR LINE SPLI
4 42 t) SHAT! d)	GAMES n) RENE DESCARTES	4 3 t) COMPUTER-GENERATED	INDEX NOW READY d) MISC
3 31 , PROC EXITS, PROGRAM AND	GARBAGECOLLECT, BAUD SWIT	1 8 t) PRAGMA IS PREPARING AN	INDEX d) MISC
4 3 t) COMPUTER-	GENERATED INDEX NOW READY	2 19 t) PRAGMA IS PREPARING AN	INDEX d) MISC
3 38 t)	GENERATING BLANK FORMS d)	3 20 t) PRAGMA IS PREPARING AN	INDEX d) MISC
3 28 t)	GENERATING MONTHLY COLUMN	3 5 d) AD v)	INFORMATION RESEARCH OF N
2 8 MING p) WIZARD c) PROGRAM	GENERATOR	4 5 d) AD v)	INFORMATION RESEARCH OF N
3 8 MING p) WIZARD c) PROGRAM	GENERATOR	3 27 CONVERSIONS, SELECT ONE,	INPUT TIME LIMIT, EDITOR
3 39 P p) AIDS c) APPLICATIONS	GENERATOR	2 14 SMAP, PART 2: FILE FORMAT	INPUT d) ARTICLE
3 47 TA p) HAL c) APPLICATIONS	GENERATOR	3 15 P, PART 3: REMAINING FILE	INPUT d) ARTICLE
4 7 P p) AIDS c) APPLICATIONS	GENERATOR	2 27 PART 2: BILL OF MATERIAL	INPUT d) ARTICLE k) ENGIN
4 41 MING p) WIZARD c) PROGRAM	GENERATOR	1 27 ISH PRINT LIMITING, BASIC	INPUT/OUTPUT, SYSTEM RELE
4 48 TA p) HAL c) APPLICATIONS	GENERATOR	2 41 d) AD v)	INTERACTIVE SYSTEMS p) CO
4 29 JEANINE CHRISTIANO, DICK	GOULD	3 I d) AD v)	INTERACTIVE SYSTEMS p) CO
3 44 PRISES p) ACCU-PLOT II c)	GRAPHICS SOFTWARE	4 31 d) AD v)	INTERACTIVE SYSTEMS p) CO
4 44 PRISES p) ACCU-PLOT II c)	GRAPHICS SOFTWARE	4 I d) AD v)	INTERACTIVE SYSTEMS p) SC
1 43 AD v) MICRU INTL c) USER	GROUP	1 45 d) AD v)	INTERACTIVE SYSTEMS p) TC
3 27 OR DE, IDENTIFIER LENGTH,	GROUP LOCKS, ATTRIBUTE SE	2 24 t) AN	INTRODUCTION TO ENGLISH,
3 39 d) AD v) SOFTWARE	GROUP p) AIDS c) APPLICAT	3 31 t) AN	INTRODUCTION TO ENGLISH,
4 7 d) AD v) SOFTWARE	GROUP p) AIDS c) APPLICAT	4 30 t) AN	INTRODUCTION TO ENGLISH,
4 25 d) LOCAL USER	GROUPS k) DELAWARE VALLEY	2 8 d) AD v)	IRVINE COMPUTER c) PERIPH
4 35 CK SYSTEMS p) PICK POCKET	GUIDE c) DOCUMENTATION	3 33 d) AD v)	IRVINE COMPUTER c) PERIPH
3 36 DULOS d) LETTERS a) BRIAN	GULINO n) JOSEPH ZELIGS	2 35 t) 16-WAYS FROM	IRVINE d) LETTERS k) BAUD
4 3 WELCH, PETER COY, CHERYL	GUMINIK, SHIRLEY HARVEY,	2 25 RKLER, LESLIE KEENE, BILL	IRVINE, JEFF HARBOUR
3 47 d) AD v) CHRONON DATA p)	HAL c) APPLICATIONS GENER	2 22 E NUMBERS, CHECK SPOOLER,	ISTAT AND HASH-TEST, BACK
4 48 d) AD v) CHRONON DATA p)	HAL c) APPLICATIONS GENER	1 33 t) MAKING	ITEM IDENTIFIERS HASH WEL
2 16 t) GALLINSKI	HAMBURG d) USER PROFILE k	2 31 SM VS. ADDS, WHERE LINKS,	ITEM LOCKS, CHARGEBACK AL
2 25 KEENE, BILL IRVINE, JEFF	HARBOUR	4 3 NS, BRUCE SLAWINSKI, FRAN	JACOBSON, TJ LEIFIELD, JA
4 30 T AIDE d) LETTERS a) JACK	HARDMAN n) MIKE ROSSETTI,	2 24 CTION TO ENGLISH, PART 1:	JARGON d) ARTICLE
2 43 d) SUBSCRIBERS n) JOHN	HARDMAN, WARD VUILLENOT,	4 18 CORRECTING TYPOS a) MARK	JOHNSON
1 44 v) SHEBASTA ASSOC c) USED	HARDWARE	3 5 YORK p) REALITY TECHNICAL	JOURNAL c) DOCUMENTATION
1 48 ATIONS CONTROLLER c) PORT	HARDWARE	4 5 YORK p) REALITY TECHNICAL	JOURNAL c) DOCUMENTATION
2 8 NE COMPUTER c) PERIPHERAL	HARDWARE	4 17 t) LEFT VS. RIGHT	JUSTIFICATION d) BENCHMAR
2 8 LINE OPERATOR c) PRINTER	HARDWARE	3 26 t)	JUSTIFYING RAGGED OUTPUT
3 8 AD v) EASTCOMP II c) USED	HARDWARE	2 40 RRIS, THOMAS DOWLING, DON	KATZMAN, PEGGY MORRISON
3 11 d) AD v) SHEBESTA c) USED	HARDWARE	1 43 d) AD v)	KDK ENTERPRISES p) GAAP c

2 4	d) AD v)	KDK ENTERPRISES p) GAAP c	3 17	FOODS d) USER PROFILE k)	MICRODATA TO ADDS CONVERS
3 41	d) AD v)	KDK ENTERPRISES p) GAAP c	4 14	Y BANK d) USER PROFILE k)	MICRODATA TO PRIME CONVER
4 8	d) AD v)	KDK ENTERPRISES p) GAAP c	4 18	E k) COSMOS VS. PRIME VS.	MICRODATA VS. ADDS n) ROG
2 35	S k) BAUD RATES a) DENNIS	KEATS	1 47	d) AD v)	MICRODATA p) REALITY c) C
2 25	J MICHAEL VERKLER, LESLIE	KEENE, BILL IRVINE, JEFF	1 43	d) AD v)	MICRU INTL c) USER GROUP
2 16	EL CONVERSION, CPA n) JOE	KEMPTER	2 43	t) DID YOU	MISS PRAGMA #1? d) MISC
3 17	IBUTOR, WIZARD n) STEPHEN	KOWARSKY	1 26	t) HOW TO FLAG	MISSING CHECKS d) ARTICLE
3 35	NS a) AL RIELAND n) STEVE	KOWARSKY	4 41	ER SYSTEMS p) FIXED ASSET	MODULE c) ACCOUNTING PACK
3 25	TE, OCONV, FIELD n) STEVE	KOWARSKY, ROGER HARPEL	2 6	t) A	MODULO SETTING PROGRAM d)
2 43	AN, WARD VUILLENOT, SERGE	KRISIUK, LEO FOHL	1 29	t) COMPUTING	MODULOS WITH ENGLISH d) A
1 27	T, SYSTEM RELEASES, SORT-	LABEL, ON ELSE, BLOCK-CON	3 36	t) MORE ON COMPUTING	MODULOS d) LETTERS a) BRI
1 9	FOR RENUMBERING STATEMENT	LABELS d) UTILITIES	2 35	t) COMPUTING	MODULOS d) LETTERS a) JOS
2 17	VENTING LOGONS, STATEMENT	LABELS, FILE ACCESS, POWE	1 18	t)	MOLETRON d) USER PROFILE
4 14	E CONVERSION n) ALEXANDER	LANGE	3 33	SOC p) SAM c) PERFORMANCE	MONITOR
2 20	TE d) BENCHMARKS n) DAVIN	LAWSON, KENT SULPRISIO	4 41	SOC p) SAM c) PERFORMANCE	MONITOR
3 40	t) ANIMAL, A PROGRAM THAT	LEARNS d) GAMES k) BINARY	3 28	t) GENERATING	MONTHLY COLUMN HEADINGS d
4 17	t)	LEFT VS. RIGHT JUSTIFICAT	3 25	t)	MORE BASIC COMPARISONS d)
2 10	AN OLDS, BILL VAUGHN, JIM	LEGGETT	3 31	CTION TO ENGLISH, PART 2:	MORE COMMANDS d) ARTICLE
4 3	WINSKI, FRAN JACOBSON, TJ	LEIFIELD, JANICE BARTLETT	1 21	t)	MORE MEMORY, FEWER READS
3 27	NS, EDITOR DE, IDENTIFIER	LENGTH, GROUP LOCKS, ATTR	3 36	t)	MORE ON COMPUTING MODULOS
1 35	d)	LETTERS	2 40	CRIBERS n) STEVEN DAVIES-	MORRIS, THOMAS DOWLING, D
3 36	E ON COMPUTING MODULOS d)	LETTERS a) BRIAN GULINO n	2 40	MLING, DON KATZMAN, PEGGY	MORRISON
4 30	t) ANOTHER EDIT AIDE d)	LETTERS a) JACK HARDMAN n	1 34	d) COMPUTER ROOM k) TAPE	MOUNTING/DISMOUNTING
2 35	t) COMPUTING MODULOS d)	LETTERS a) JOSEPH ZELIGS	1 15	PARTS FILE d) ARTICLE k)	MRP, SCREENPRO, ENGINEERI
3 36	t) ACCU/PLOT UPGRADES d)	LETTERS a) MIKE SCHELLENB	4 29	WHITE, JIM RENALDS, TERRY	MULHERN, DENNIS BROWN, RO
2 35	t) 16-WAYS FROM IRVINE d)	LETTERS k) BAUD RATES a)	3 7	d) AD v)	MULTIDATA p) FIXED ASSET
3 35	ME WISHES ARE ANSWERED d)	LETTERS k) SLEEP, EDITOR	4 33	d) AD v)	MULTIDATA p) FIXED ASSET
4 25	IBLEY, HENRY HERMAN, LISA	LEVSEN	1 27	DITOR PRESTORE, SCREENPRO	MULTIVALUES, BASIC INCLUD
2 3	t) WE HAVE	LIFTOFF d) EDITORIAL k) R	2 22	ING ASSIGNMENT, SCREENPRO	MULTIVALUES, PAGE EDITOR,
3 27	S, SELECT ONE, INPUT TIME	LIMIT, EDITOR F	3 8	n) GEORGE LOPATIN, ROBERT	MURRAY, JOHN VETTER, JOEL
1 27	GON LIMITS, ENGLISH PRINT	LIMITING, BASIC INPUT/OUT	2 3	R PREFERENCES, SUBSCRIBER	NAMES, SPREADSHEETS, SUBM
1 27	TOR LINE SPLITTING, LOGON	LIMITS, ENGLISH PRINT LIM	2 45	ERS n) ARTHUR HUBER, MARK	NELSON, MALIA CULBRETH, J
3 27	SPOOLER ACCOUNTING, LOCK	LIMITS, SLOW COMMANDS, MF	4 41	ASSOC p) THE SIDEKICK c)	NETWORK HARDWARE/SOFTWARE
2 22	R, CORRELATIVE CALLS, ON-	LINE DOCUMENTATION, ASSER	1 46	AD v) COSMOS p) R/NET c)	NETWORK SOFTWARE
2 8	CONSULTANTS p) AUTOMATIC	LINE OPERATOR c) PRINTER	3 37	AD v) COSMOS p) R/NET c)	NETWORK SOFTWARE
1 27	ES, BASIC INCLUDE, EDITOR	LINE SPLITTING, LOGON LIM	4 43	AD v) COSMOS p) R/NET c)	NETWORK SOFTWARE
3 35	DITOR PRESTORE, SPLITTING	LINES, PREVENTING LOGONS	3 5	) INFORMATION RESEARCH OF	NEW YORK p) REALITY TECHN
2 31	CE, PRISM VS. ADDS, WHERE	LINKS, ITEM LOCKS, CHARGE	4 5	) INFORMATION RESEARCH OF	NEW YORK p) REALITY TECHN
3 12	t)	LIST.POINTERS: A PROC FOR	1 46	t) IN THE	NEXT PRAGMA d) MISC
4 26	t) AVOIDING SAVED	LISTS WITH PQ-RESELECT d)	3 8	MURRAY, JOHN VETTER, JOEL	NIRENBERG, BRUCE CORBETT,
3 31	d) QUERIES k)	LISTU, PROC EXITS, PROGRA	1 15	t) VANILLA, THE	NO-FRILLS MANUFACTURING S
4 25	d)	LOCAL USER GROUPS k) DELA	2 27	t) VANILLA, THE	NO-FRILLS MANUFACTURING S
2 20	t) LOOP VS.	LOCATE d) BENCHMARKS n) D	3 30	t) VANILLA, THE	NO-FRILLS MANUFACTURING S
2 22	E, UNDOCUMENTED FEATURES,	LOCATE, EDITOR U, PQ VS.	4 32	t) VANILLA, THE	NO-FRILLS MANUFACTURING S
3 25	NS d) BENCHMARKS k) LOOP,	LOCATE, OCONV, FIELD n) S	4 25	DELAWARE VALLEY, ARIZONA,	NORTHERN CALIFORNIA n) BO
3 27	INGS, SPOOLER ACCOUNTING,	LOCK LIMITS, SLOW COMMAND	4 18	d) WISH LIST k) COPY	NULL SUPPRESS, DEFAULT AT
3 27	IDENTIFIER LENGTH, GROUP	LOCKS, ATTRIBUTE SECURITY	2 31	GEBACK ALGORITHM, RELEASE	NUMBERS, BREAK-ON, SYSPRO
2 31	ADDS, WHERE LINKS, ITEM	LOCKS, CHARGEBACK ALGORIT	2 22	DITOR U, PQ VS. PGN, PAGE	NUMBERS, CHECK SPOOLER, I
1 27	FILE ACCESS, POWER-FAILS,	LOGON HISTORY	4 12	T 1 d) UTILITIES k) BASIC	OBJECT CODE
1 27	E, EDITOR LINE SPLITTING,	LOGON LIMITS, ENGLISH PRI	3 25	NCHMARKS k) LOOP, LOCATE,	OCONV, FIELD n) STEVE KOW
3 35	LITTING LINES, PREVENTING	LOGONS a) AL RIELAND n) S	2 10	IBERS n) JUDIE RANDS, DAN	OLDS, BILL VAUGHN, JIM LE
1 27	S, BAUD RATES, PREVENTING	LOGONS, STATEMENT LABELS,	1 27	TEM RELEASES, SORT-LABEL,	ON ELSE, BLOCK-CONVERT, F
2 20	t)	LOOP VS. LOCATE d) BENCHM	2 22	ISTER, CORRELATIVE CALLS,	ON-LINE DOCUMENTATION, AS
3 25	PARISONS d) BENCHMARKS k)	LOOP, LOCATE, OCONV, FIEL	4 2	) COSMOS p) REVELATION c)	OPERATING SYSTEM
3 8	d) SUBSCRIBERS n) GEORGE	LOPATIN, ROBERT MURRAY, J	2 8	ULTANTS p) AUTOMATIC LINE	OPERATOR c) PRINTER HARDW
3 7	MULTIDATA p) FIXED ASSET	MANAGEMENT SYSTEM c) ACCO	4 25	d) ARTICLE k) SEMI-COLON	OPERATOR n) ED SHEEHAN
4 33	MULTIDATA p) FIXED ASSET	MANAGEMENT SYSTEM c) ACCO	4 32	SYSTEM, PART 4: PURCHASE	ORDER ENTRY d) ARTICLE
4 34	d) AD v) PAULA	MANNILLO p) MICRODATA REA	4 18	DDS n) ROGER HARPEL, DAVE	OSTBY, ALLISON HARTMANN,
1 41	p) CONSULTING SERVICES c)	MANUFACTURING	3 26	t) JUSTIFYING RAGGED	OUTPUT d) ARTICLE k) ATTR
2 48	p) CONSULTING SERVICES c)	MANUFACTURING	2 22	, EDITOR DE, FILE ENGLISH	OUTPUT, CONTIGUOUS OUTPUT
3 33	p) CONSULTING SERVICES c)	MANUFACTURING	2 22	NGLISH OUTPUT, CONTIGUOUS	OUTPUT, STRING ASSIGNMENT
3 48	p) CONSULTING SERVICES c)	MANUFACTURING	1 27	NT LIMITING, BASIC INPUT/	OUTPUT, SYSTEM RELEASES,
1 15	t) VANILLA, THE NO-FRILLS	MANUFACTURING SYSTEM, PAR	4 14	t)	PACIFIC VALLEY BANK d) US
2 27	t) VANILLA, THE NO-FRILLS	MANUFACTURING SYSTEM, PAR	1 43	SES p) GAAP c) ACCOUNTING	PACKAGE
3 30	t) VANILLA, THE NO-FRILLS	MANUFACTURING SYSTEM, PAR	2 4	SES p) GAAP c) ACCOUNTING	PACKAGE
4 32	t) VANILLA, THE NO-FRILLS	MANUFACTURING SYSTEM, PAR	3 7	MENT SYSTEM c) ACCOUNTING	PACKAGE
1 18	ECTRON d) USER PROFILE k)	MANUFACTURING, MICRODATA	3 41	SES p) GAAP c) ACCOUNTING	PACKAGE
4 30	n) MIKE ROSSETTI, HOWARD	MARKS, BRIAN HILL, MARK H	4 8	SES p) GAAP c) ACCOUNTING	PACKAGE
4 3	ONHAM, ANNA CARSEN, STEVE	MATTSON, TOM WELCH, PETER	4 33	MENT SYSTEM c) ACCOUNTING	PACKAGE
3 13	n) ROBERT SCHMIDT, PETER	MCOMBER, KIRBY PLUMLEE	4 41	SSET MODULE c) ACCOUNTING	PACKAGE
1 21	t) MORE	MEMORY, FEWER READS d) BE	2 22	T, SCREENPRO MULTIVALUES,	PAGE EDITOR, E REGISTER,
3 27	CK LIMITS, SLOW COMMANDS,	MF CONVERSIONS, SELECT ON	2 22	TE, EDITOR U, PQ VS. PGN,	PAGE NUMBERS, CHECK SPOOL
4 34	) AD v) PAULA MANNILLO p)	MICRODATA REALITY c) USED	2 22	DITOR B, PQ-COMPIL STON,	PAGE, UNDOCUMENTED FEATUR
3 33	d) AD v) ALPHATEST p)	MICRODATA REALITY c) WANT	3 34	t) CONVERTING	PAINT TO PROGRAMS d) ARTI
1 18	PROFILE k) MANUFACTURING,	MICRODATA REALITY n) JUDI	4 12	t) UNCOMPILING,	PART 1 d) UTILITIES k) BA

1 22	t) SYSMAP,	PART 1: A CROSS-REFERENCE	3 12	t) LIST.POINTERS: A	PROC FOR CROSS-REFERENCIN
2 24	INTRODUCTION TO ENGLISH,	PART 1: JARGON d) ARTICLE	1 30	ING ENGLISH HEADINGS WITH	PROC d) COMMAND FILES k)
1 15	LLS MANUFACTURING SYSTEM,	PART 1: THE PARTS FILE d)	4 1	S p) SCREEN-GEN c) SCREEN	PROCESSOR
2 27	LLS MANUFACTURING SYSTEM,	PART 2: BILL OF MATERIAL	4 6	SYSMAP, PART 4: DICTS AND	PROCS d) ARTICLE
2 14	t) SYSMAP,	PART 2: FILE FORMAT INPUT	3 3	SURVEY, IDBMA DIRECTORY,	PRODUCT EVALUATION, CONFI
3 31	INTRODUCTION TO ENGLISH,	PART 2: MORE COMMANDS d)	2 11	t) TRIM.DELIM AND	PROFILE d) UTILITIES k) F
4 30	INTRODUCTION TO ENGLISH,	PART 3: FINDING FILES d)	1 18	t) MOLECTRON d) USER	PROFILE k) MANUFACTURING,
3 30	LLS MANUFACTURING SYSTEM,	PART 3: PURCHASING d) ART	3 17	BOW NATURAL FOODS d) USER	PROFILE k) MICRODATA TO A
3 15	t) SYSMAP,	PART 3: REMAINING FILE IN	4 14	CIFIC VALLEY BANK d) USER	PROFILE k) MICRODATA TO P
4 6	t) SYSMAP,	PART 4: DICTS AND PROCS d	2 16	GALLINSKI HAMBURG d) USER	PROFILE k) REALITY TO SEQ
4 32	LLS MANUFACTURING SYSTEM,	PART 4: PURCHASE ORDER EN	3 31	IES k) LISTU, PROC EXITS,	PROGRAM AND GARBAGECOLLEC
1 15	URING SYSTEM, PART 1: THE	PARTS FILE d) ARTICLE k)	1 22	ENCE SYSTEM d) ARTICLE k)	PROGRAM DOCUMENTATION
2 22	d) WISH LIST k)	PASSWORDS, DISPLAY STATEM	1 9	t) RENUMBER: A	PROGRAM FOR RENUMBERING S
1 36	t)	PATCHING THE EXIT PROBLEM	2 8	PROGRAMMING p) WIZARD c)	PROGRAM GENERATOR
3 33	) BURNS & ASSOC p) SAM c)	PERFORMANCE MONITOR	3 8	PROGRAMMING p) WIZARD c)	PROGRAM GENERATOR
4 41	) BURNS & ASSOC p) SAM c)	PERFORMANCE MONITOR	4 41	PROGRAMMING p) WIZARD c)	PROGRAM GENERATOR
2 8	AD v) IRVINE COMPUTER c)	PERIPHERAL HARDWARE	3 40	t) ANIMAL, A	PROGRAM THAT LEARNS d) GA
3 33	AD v) IRVINE COMPUTER c)	PERIPHERAL HARDWARE	2 6	t) A MODULO SETTING	PROGRAM d) ARTICLE
3 8	COMPUTER DISTRIBUTORS c)	PERSONNEL WANTED	2 38	CURITY AND THE DATA/BASIC	PROGRAMMER d) ARTICLE
1 3	k) PRAGMATISM n) RICHARD	PICK	2 8	d) AD v) AUTOMATIC	PROGRAMMING p) WIZARD c)
4 35	d) AD v) PICK SYSTEMS p)	PICK POCKET GUIDE c) DOCU	3 8	d) AD v) AUTOMATIC	PROGRAMMING p) WIZARD c)
3 33	d) AD v)	PICK SYSTEMS c) CONTEST	4 41	d) AD v) AUTOMATIC	PROGRAMMING p) WIZARD c)
4 35	d) AD v)	PICK SYSTEMS p) PICK POCK	3 34	t) CONVERTING PAINT TO	PROGRAMS d) ARTICLE k) SC
4 25	ER, JODI HILGENBERG, MARK	PICK, MIKE SIBLEY, HENRY	4 32	FACTURING SYSTEM, PART 4:	PURCHASE ORDER ENTRY d) A
4 29	OF HARDWARE THAT SUPPORTS	PICK-STYLE SOFTWARE d) AR	3 30	FACTURING SYSTEM, PART 3:	PURCHASING d) ARTICLE
3 13	DT, PETER MCCOMBER, KIRBY	PLUMLEE	3 12	ROC FOR CROSS-REFERENCING	Q-POINTERS d) UTILITIES
4 35	D v) PICK SYSTEMS p) PICK	POCKET GUIDE c) DOCUMENTA	1 32	d)	QUERIES k) BINARY AND FIL
3 12	FOR CROSS-REFERENCING Q-	POINTERS d) UTILITIES	4 27	d)	QUERIES k) IMPROVING THRO
3 12	t) LIST,	POINTERS: A PROC FOR CROS	3 31	d)	QUERIES k) LISTU, PROC EX
1 48	MUNICATIONS CONTROLLER c)	PORT HARDWARE	2 31	d)	QUERIES k) WORKSPACE, HOL
2 36	t) A SWITCHBOX FOR 32	PORTS d) ARTICLE k) SECUR	1 46	d) AD v) COSMOS p)	R/NET c) NETWORK SOFTWARE
1 27	MENT LABELS, FILE ACCESS,	POWER-FAILS, LOGON HISTOR	3 37	d) AD v) COSMOS p)	R/NET c) NETWORK SOFTWARE
2 22	ATURES, LOCATE, EDITOR U,	PQ VS. PQN, PAGE NUMBERS,	4 43	d) AD v) COSMOS p)	R/NET c) NETWORK SOFTWARE
2 22	EENPRO COMMAND, EDITOR B,	PQ-COMPILE STON, PAGE, UN	3 26	t) JUSTIFYING	RAGGED OUTPUT d) ARTICLE
4 26	AVOIDING SAVED LISTS WITH	PQ-RESELECT d) COMMAND FI	3 17	t)	RAINBOW NATURAL FOODS d)
2 22	LOCATE, EDITOR U, PQ VS.	PQN, PAGE NUMBERS, CHECK	1 18	ICRODATA REALITY n) JUDIE	RANDS
2 43	t) DID YOU MISS	PRAGMA #1? d) MISC	2 10	d) SUBSCRIBERS n) JUDIE	RANDS, DAN OLDS, BILL VAU
3 3	t) IS	PRAGMA A RARE MEDIUM, MEL	2 35	IRVINE d) LETTERS k) BAUD	RATES a) DENNIS KEATS
1 3	t)	PRAGMA IS LOOKING FOR SUB	1 27	CURITY, SORT BREAKS, BAUD	RATES, PREVENTING LOGONS,
2 3	t)	PRAGMA IS LOOKING FOR SUB	2 3	E LIFTOFF d) EDITORIAL k)	READER PREFERENCES, SUBSC
1 8	t)	PRAGMA IS PREPARING AN IN	1 2	SUBMITTALS, ADVERTISING,	READER SERVICES
2 19	t)	PRAGMA IS PREPARING AN IN	2 2	SUBMITTALS, ADVERTISING,	READER SERVICES
3 20	t)	PRAGMA IS PREPARING AN IN	3 4	SUBMITTALS, ADVERTISING,	READER SERVICES
1 43	d) AD v) SEMAPHORE p)	PRAGMA c) ADVERTISING SPA	4 4	SUBMITTALS, ADVERTISING,	READER SERVICES
2 8	d) AD v) SEMAPHORE p)	PRAGMA c) ADVERTISING SPA	4 3	Y SAYS... d) EDITORIAL k)	READER SURVEY, AVAILABLE
3 30	d) AD v) SEMAPHORE p)	PRAGMA c) ADVERTISING SPA	3 3	ELL DONE? d) EDITORIAL k)	READER SURVEY, IDBMA DIRE
1 3	d) AD v) SEMAPHORE p)	PRAGMA c) SUBSCRIPTION	1 21	t) MORE MEMORY, FEWER	READS d) BENCHMARKS
1 45	d) AD v) SEMAPHORE p)	PRAGMA c) SUBSCRIPTION	3 5	N RESEARCH OF NEW YORK p)	REALITY TECHNICAL JOURNAL
1 46	d) AD v) SEMAPHORE p)	PRAGMA c) SUBSCRIPTION	4 5	N RESEARCH OF NEW YORK p)	REALITY TECHNICAL JOURNAL
2 3	d) AD v) SEMAPHORE p)	PRAGMA c) SUBSCRIPTION	2 16	AMBURG d) USER PROFILE k)	REALITY TO SEQUEL CONVERS
2 33	d) AD v) SEMAPHORE p)	PRAGMA c) SUBSCRIPTION	1 47	d) AD v) MICRODATA p)	REALITY c) COMPUTER SYSTE
2 43	d) AD v) SEMAPHORE p)	PRAGMA c) SUBSCRIPTION	4 34	ULA MANNILLO p) MICRODATA	REALITY c) USED HARDWARE
3 37	d) AD v) SEMAPHORE p)	PRAGMA c) SUBSCRIPTION	3 33	v) ALPHATEST p) MICRODATA	REALITY c) WANTED TO BUY
4 24	d) AD v) SEMAPHORE p)	PRAGMA c) SUBSCRIPTION	1 18	MANUFACTURING, MICRODATA	REALITY n) JUDIE RANDS
1 3	t) WELCOME TO	PRAGMA d) EDITORIAL k) PR	3 32	COMPUTER ROOM k) DISASTER	RECOVERY
1 46	t) IN THE NEXT	PRAGMA d) MISC	1 22	SYSMAP, PART 1: A CROSS-	REFERENCE SYSTEM d) ARTIC
1 3	TO PRAGMA d) EDITORIAL k)	PRAGMATISM n) RICHARD PIC	3 12	INTERS: A PROC FOR CROSS-	REFERENCING Q-POINTERS d)
2 3	FF d) EDITORIAL k) READER	PREFERENCES, SUBSCRIBER N	3 37	d) AD v) ZENTEC p)	REFLEX c) USED HARDWARE
1 8	t) PRAGMA IS	PREPARING AN INDEX d) MIS	4 34	d) AD v) ZENTEC p)	REFLEX c) USED HARDWARE
2 19	t) PRAGMA IS	PREPARING AN INDEX d) MIS	2 22	LTIVALUES, PAGE EDITOR, E	REGISTER, CORRELATIVE CAL
3 20	t) PRAGMA IS	PREPARING AN INDEX d) MIS	4 25	IA n) BOB CLEARFIELD, JIM	REILLY, JIM CATES, SHELLEY
1 27	ME FAULTS, ERRMSG, EDITOR	PRESTORE, SCREENPRO MULTI	2 31	KS, CHARGEBACK ALGORITHM,	RELEASE NUMBERS, BREAK-ON
3 35	LETTERS k) SLEEP, EDITOR	PRESTORE, SPLITTING LINES	1 27	ASIC INPUT/OUTPUT, SYSTEM	RELEASES, SORT-LABEL, ON
3 35	RESTORE, SPLITTING LINES,	PREVENTING LOGONS a) AL R	3 15	t) SYSMAP, PART 3:	REMAINING FILE INPUT d) A
1 27	SORT BREAKS, BAUD RATES,	PREVENTING LOGONS, STATEM	4 29	TICLE n) ROBIN WHITE, JIM	RENALDS, TERRY MULHERN, D
4 14	R PROFILE k) MICRODATA TO	PRIME CONVERSION n) ALEXA	1 9	t)	RENUMBER: A PROGRAM FOR R
4 18	d) ARTICLE k) COSMOS VS.	PRIME VS. MICRODATA VS. A	1 9	) RENUMBER: A PROGRAM FOR	RENUMBERING STATEMENT LAB
1 27	NG, LOGON LIMITS, ENGLISH	PRINT LIMITING, BASIC INP	1 45	E SYSTEMS p) TCL-AUDITOR,	REPORT-GEN, SCREEN-GEN c)
2 8	UTOMATIC LINE OPERATOR c)	PRINTER HARDWARE	4 27	t) SELF-DOCUMENTING	REPORTS d) ARTICLE k) ENG
4 29	t)	PRINTER TRADE-OFFS d) COM	4 26	BING SAVED LISTS WITH PQ-	RESELECT d) COMMAND FILES
2 31	TY, AVAILABLE DISK SPACE,	PRISM VS. ADDS, WHERE LIN	1 32	k) BINARY AND FILE SAVE/	RESTORE, ALL, 4.1 SPOOLER
1 36	t) PATCHING THE EXIT	PROBLEM IN 3.2 SCREENPRO	4 2	d) AD v) COSMOS p)	REVELATION c) OPERATING S
3 6	DES d) ARTICLE k) DEFAULT	PROC ARGUMENTS a) MICHAEL	3 35	, PREVENTING LOGONS a) AL	RIELAND n) STEVE KOWARSKY
2 26	t)	PROC CONVERSIONS d) COMMA	4 17	t) LEFT VS.	RIGHT JUSTIFICATION d) BE
3 31	d) QUERIES k) LISTU,	PROC EXITS, PROGRAM AND G	3 6	PROC ARGUMENTS a) MICHAEL	ROSSETTI

4 30	S a)	JACK HARDMAN n)	MIKE	ROSSETTI, HOWARD MARKS, B	4 44	ACCU-PLOT II c)	GRAPHICS	SOFTWARE		
3 33	d)	AD v)	BURNS & ASSOC p)	SAM c)	PERFORMANCE MONITO	3 39	d)	AD v)	SOFTWARE GROUP p)	AIDS c)
4 41	d)	AD v)	BURNS & ASSOC p)	SAM c)	PERFORMANCE MONITO	4 7	d)	AD v)	SOFTWARE GROUP p)	AIDS c)
1 32	UERIES k)	BINARY AND FILE	SAVED LISTS WITH PQ-RESEL	4 15	REPORT-GEN, SCREEN-GEN c)	SOFTWARE TOOLS				
4 26	t)	AVOIDING	SCHLLENBACH	4 29	THAT SUPPORTS PICK-STYLE	SOFTWARE d)	ARTICLE n)	RO		
3 36	GRADES d)	LETTERS a)	MIKE	1 27	K-CONVERT, FILE SECURITY,	SORT BREAKS, BAUD RATES,				
3 13	d)	SUBSCRIBERS n)	ROBERT	1 27	/OUTPUT, SYSTEM RELEASES,	SORT-LABEL, ON ELSE, BLOC				
4 1	I	SYSTEMS p)	SCREEN-GEN c)	4 17	FICATION d)	BENCHMARKS k)	SORTING			
4 1	v)	INTERACTIVE SYSTEMS p)	SCREEN-GEN c)	2 31	RS, BREAK-ON, SYSPROG VS.	SP, CREATING FILES				
1 45	TCL-AUDITOR, REPORT-GEN,	SCREEN-GEN c)	SOFTWARE TO	2 31	SECURITY, AVAILABLE DISK	SPACE, PRISM VS. ADDS, WH				
3 34	TO PROGRAMS d)	ARTICLE k)	SCREENPRO	4 29	LL, SHIRLEY STOUGH, LINDA	SPELMAN, LEE ADAMSON, JEA				
2 22	OCUMENTATION, ASSERTIONS,	SCREENPRO COMMAND, EDITOR		3 35	) SLEEP, EDITOR PRESTORE,	SPLITTING LINES, PREVENTI				
2 22	WORDS, DISPLAY STATEMENT,	SCREENPRO CONVERSIONS, ED		1 27	ASIC INCLUDE, EDITOR LINE	SPLITTING, LOGON LIMITS,				
1 27	ERRMSG, EDITOR PRESTORE,	SCREENPRO MULTIVALUES, BA		1 32	LE SAVE/RESTORE, ALL, 4.1	SPOOLER				
2 22	UTPUT, STRING ASSIGNMENT,	SCREENPRO MULTIVALUES, PA		3 27	CURITY, ENGLISH HEADINGS,	SPOOLER ACCOUNTING, LOCK				
1 36	G THE EXIT PROBLEM IN 3.2	SCREENPRO d)	ARTICLE k)	2 22	PQN, PAGE NUMBERS, CHECK	SPOOLER, ISTAT AND HASH-T				
1 15	S FILE d)	ARTICLE k)	MRP,	1 24	v)	SEMAPHORE p)	CARGO c)	SPREADSHEET		
2 36	OR 32 PORTS d)	ARTICLE k)	SECURITY	2 5	) AUROTECH p)	AUROPLAN c)	SPREADSHEET			
2 38	t)	SECURITY AND THE DATA/BAS		2 41	SYSTEMS p)	COMPU-SHEET c)	SPREADSHEET			
2 31	P AND BLAST, ERRMSG [13],	SECURITY, AVAILABLE DISK		2 46	v)	SEMAPHORE p)	CARGO c)	SPREADSHEET		
3 27	H, GROUP LOCKS, ATTRIBUTE	SECURITY, ENGLISH HEADING		3 2	) AUROTECH p)	AUROPLAN c)	SPREADSHEET			
1 27	ELSE, BLOCK-CONVERT, FILE	SECURITY, SORT BREAKS, BA		3 47	v)	SEMAPHORE p)	CARGO c)	SPREADSHEET		
4 26	ELECT d)	COMMAND FILES k)	SELECT	3 1	SYSTEMS p)	COMPU-SHEET c)	SPREADSHEET			
3 27	COMMANDS, MF CONVERSIONS,	SELECT ONE, INPUT TIME LI		4 31	SYSTEMS p)	COMPU-SHEET c)	SPREADSHEET			
4 27	t)	SELF-DOCUMENTING REPORTS		4 47	) AUROTECH p)	AUROPLAN c)	SPREADSHEET			
1 24	d)	AD v)	SEMAPHORE p)	2 3	RENCE, SUBSCRIBER NAMES,	SPREADSHEETS, SUBMITTALS				
2 46	d)	AD v)	SEMAPHORE p)	1 9	A PROGRAM FOR RENUMBERING	STATEMENT LABELS d)	UTILI			
3 47	d)	AD v)	SEMAPHORE p)	1 27	RATES, PREVENTING LOGONS,	STATEMENT LABELS, FILE AC				
1 41	d)	AD v)	SEMAPHORE p)	2 22	IST k)	PASSWORDS, DISPLAY	STATEMENT, SCREENPRO CONV			
2 48	d)	AD v)	SEMAPHORE p)	2 22	AND, EDITOR B, PQ-COMPILE	STON, PAGE, UNDOCUMENTED				
3 48	d)	AD v)	SEMAPHORE p)	4 29	PEL, PAUL DAHILL, SHIRLEY	STOUGH, LINDA SPELMAN, LE				
1 43	d)	AD v)	SEMAPHORE p)	2 22	UTPUT, CONTIGUOUS OUTPUT,	STRING ASSIGNMENT, SCREEN				
2 8	d)	AD v)	SEMAPHORE p)	2 3	IBER NAMES, SPREADSHEETS,	SUBMITTALS				
3 30	d)	AD v)	SEMAPHORE p)	1 3	t)	PRAGMA IS LOOKING FOR	SUBMITTALS d)	MISC		
1 3	d)	AD v)	SEMAPHORE p)	2 3	t)	PRAGMA IS LOOKING FOR	SUBMITTALS d)	MISC		
1 45	d)	AD v)	SEMAPHORE p)	1 2	d)	MISC k)	SUBSCRIPTIONS,	SUBMITTALS, ADVERTISING,		
1 46	d)	AD v)	SEMAPHORE p)	2 2	d)	MISC k)	SUBSCRIPTIONS,	SUBMITTALS, ADVERTISING,		
2 3	d)	AD v)	SEMAPHORE p)	3 4	d)	MISC k)	SUBSCRIPTIONS,	SUBMITTALS, ADVERTISING,		
2 33	d)	AD v)	SEMAPHORE p)	4 4	d)	MISC k)	SUBSCRIPTIONS,	SUBMITTALS, ADVERTISING,		
2 43	d)	AD v)	SEMAPHORE p)	2 3	AL k)	READER PREFERENCES,	SUBSCRIBER NAMES, SPREADS			
3 37	d)	AD v)	SEMAPHORE p)	2 45	d)	SUBSCRIBERS n)	ARTHUR HUB			
4 24	d)	AD v)	SEMAPHORE p)	4 24	d)	SUBSCRIBERS n)	ED SHEEHAN			
4 25	CAPABILITY d)	ARTICLE k)	SEMI-COLON OPERATOR n)	3 8	d)	SUBSCRIBERS n)	GEORGE LOP			
2 16	SER PROFILE k)	REALITY TO	SEQUEL CONVERSION, CPA n)	2 43	d)	SUBSCRIBERS n)	JOHN HARDM			
1 37	PUTER DISTRIBUTORS p)	IDM	SERIES/1 c)	2 10	d)	SUBSCRIBERS n)	JUDIE RAND			
1 2	TALS, ADVERTISING, READER	SERVICES		2 25	d)	SUBSCRIBERS n)	MICHAEL VE			
2 2	TALS, ADVERTISING, READER	SERVICES		2 18	d)	SUBSCRIBERS n)	MIKE BLUE,			
3 4	TALS, ADVERTISING, READER	SERVICES		3 13	d)	SUBSCRIBERS n)	ROBERT SCH			
4 4	TALS, ADVERTISING, READER	SERVICES		2 40	d)	SUBSCRIBERS n)	STEVEN DAV			
1 41	) SEMAPHORE p)	CONSULTING	SERVICES c)	1 3	v)	SEMAPHORE p)	PRAGMA c)	SUBSCRIPTION		
2 48	) SEMAPHORE p)	CONSULTING	SERVICES c)	1 45	v)	SEMAPHORE p)	PRAGMA c)	SUBSCRIPTION		
3 33	D v)	ADDEPT p)	CONSULTING	1 46	v)	SEMAPHORE p)	PRAGMA c)	SUBSCRIPTION		
3 48	) SEMAPHORE p)	CONSULTING	SERVICES c)	2 3	v)	SEMAPHORE p)	PRAGMA c)	SUBSCRIPTION		
2 6	t)	A MODULO	SETTING PROGRAM d)	2 33	v)	SEMAPHORE p)	PRAGMA c)	SUBSCRIPTION		
3 32	t)	SHARED SITE CHECKLIST d)	SHAVE	2 43	v)	SEMAPHORE p)	PRAGMA c)	SUBSCRIPTION		
4 24	ARD USRY, BRIAN EHLERS, W	SHEBASTA ASSOC c)	USED HA	3 37	v)	SEMAPHORE p)	PRAGMA c)	SUBSCRIPTION		
1 44	d)	AD v)	SHEBASTA c)	4 24	v)	SEMAPHORE p)	PRAGMA c)	SUBSCRIPTION		
3 11	d)	AD v)	SHEBASTA c)	1 2	d)	MISC k)	SUBSCRIPTIONS, SUBMITTALS			
4 11	d)	AD v)	SHEBASTA c)	2 2	d)	MISC k)	SUBSCRIPTIONS, SUBMITTALS			
4 25	SEMI-COLON OPERATOR n)	ED	SHEEHAN	3 4	d)	MISC k)	SUBSCRIPTIONS, SUBMITTALS			
4 24	d)	SUBSCRIBERS n)	ED	4 4	d)	MISC k)	SUBSCRIPTIONS, SUBMITTALS			
1 38	t)	THE	SHELL GAME d)	2 20	RKS n)	DAVIN LAWSON, KENT	SULPRISIO			
4 25	LGEBERG, MARK PICK, MIKE	SIBLEY, HENRY HERMAN, LIS		4 29	A SURVEY OF HARDWARE THAT	SUPPORTS PICK-STYLE SOFTW				
4 41	D v)	ANSDATA ASSOC p)	THE	4 18	d)	WISH LIST k)	COPY NULL	SUPPRESS, DEFAULT ATTRIBU		
4 3	, PHIL HUNTINGDON, JOSEPH	SIROTA, WESLEY WOODLAND,		3 3	EVALUATION, CONFIGURATION	SURVEY				
3 32	t)	SHARED	SITE CHECKLIST d)	4 29	t)	A	SURVEY OF HARDWARE THAT S			
4 3	NG, LOUIS BELTJENS, BRUCE	SLAWINSKI, FRAN JACOBSON,		4 3	.. d)	EDITORIAL k)	READER	SURVEY, AVAILABLE HARDWAR		
1 27	d)	WISH LIST k)	SLEEP, BREAK-ON, FRAME FA	4 3	ARE SURVEY, CONFIGURATION	SURVEY, BASIC SURVEY, CON				
3 35	RE ANSWERED d)	LETTERS k)	SLEEP, EDITOR PRESTORE, S	4 3	URVEY, AVAILABLE HARDWARE	SURVEY, CONFIGURATION SUR				
3 27	ACCOUNTING, LOCK LIMITS,	SLOW COMMANDS, MF CONVERS	SOFTWARE	4 3	NFIGATION SURVEY, BASIC	SURVEY, CONTEST n)	ROGER			
1 46	OSMOS p)	R/NET c)	NETWORK	3 3	E? d)	EDITORIAL k)	READER	SURVEY, IDBMA DIRECTORY,		
3 37	OSMOS p)	R/NET c)	NETWORK	4 42	t)	SWAT! d)	GAMES n)	RENE DE		
3 44	ACCU-PLOT II c)	GRAPHICS	SOFTWARE	2 36	t)	A	SWITCHBOX FOR 32 PORTS d)			
4 41	KICK c)	NETWORK HARDWARE/	SOFTWARE	3 31	AND GARBAGECOLLECT, BAUD	SWITCHES, U508E				
4 43	OSMOS p)	R/NET c)	NETWORK							

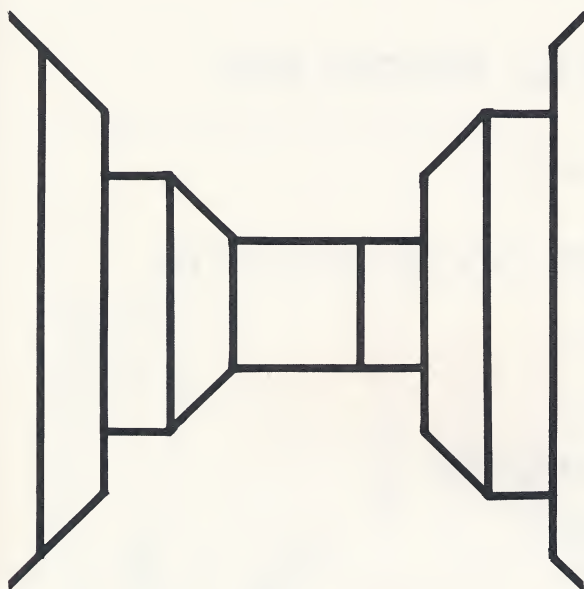
1 22	t)	SYSMAP, PART 1: A CROSS-R	2 8	AUTOMATIC PROGRAMMING p)	WIZARD c) PROGRAM GENERAT
2 14	t)	SYSMAP, PART 2: FILE FORM	3 8	AUTOMATIC PROGRAMMING p)	WIZARD c) PROGRAM GENERAT
3 15	t)	SYSMAP, PART 3: REMAINING	4 41	AUTOMATIC PROGRAMMING p)	WIZARD c) PROGRAM GENERAT
4 6	t)	SYSMAP, PART 4: DICTS AND	3 17	CONVERSION, DISTRIBUTOR,	WIZARD n) STEPHEN KOWARSK
2 31	ELSEASE NUMBERS, BREAK-ON,	SYSPROG VS. SP, CREATING	4 3	TEST n) ROGER HARPEL, RAY	WONG, LOUIS BELTJENS, BRU
1 34	d) COMPUTER ROOM k)	TAPE MOUNTING/DISMOUNTING	4 3	ON, JOSEPH SIROTA, WESLEY	WOODLAND, KAREN ANDERSON
1 45	v) INTERACTIVE SYSTEMS p)	TCL-AUDITOR, REPORT-GEN,	2 31	d) QUERIES k)	WORKSPACE, HOLD FILES, CH
3 5	CH OF NEW YORK p) REALITY	TECHNICAL JOURNAL c) DOCU	3 38	ISH CORRELATIVES a) HENRY	MUDARCZYK
4 5	CH OF NEW YORK p) REALITY	TECHNICAL JOURNAL c) DOCU	1 14	t) DERIVING	YEAR-TO-DATE COUNTS d) AR
2 22	SPOOLER, ISTAT AND HASH-	TEST, BACKSPACE	4 18	AR, CAROL TOMLINSON, DAVE	YUKLE
2 18	E BLUE, PAUL CRAVEN, ALAN	THOMPSON, LYNN ANDERSON	2 35	ULOS d) LETTERS a) JOSEPH	ZELIGS
4 27	d) QUERIES k) IMPROVING	THROUGHPUT	3 36	a) BRIAN GULINO n) JOSEPH	ZELIGS
3 27	RSIONS, SELECT ONE, INPUT	TIME LIMIT, EDITOR F	3 37	d) AD v)	ZENTEC p) REFLEX c) USED
4 18	TMANN, DAVE HARRAR, CAROL	TOMLINSON, DAVE YUKLE	4 34	d) AD v)	ZENTEC p) REFLEX c) USED
1 45	N, SCREEN-GEN c) SOFTWARE	TOOLS	1 4	t)	ZIP CODE FILE DESIGN d) A
4 29	t) PRINTER	TRADE-OFFS d) COMPUTER RO	3 35	LINES, PREVENTING LOGONS	a) AL RIELAND n) STEVE KO
2 32	t) THE TROUBLE	TREE d) COMPUTER ROOM k)	3 36	PUTING MODULOS d) LETTERS	a) BRIAN GULINO n) JOSEPH
3 40	LEARNS d) GAMES k) BINARY	TREES	2 35	d) LETTERS k) BAUD RATES	a) DENNIS KEATS
2 11	t)	TRIM.DELIM AND PROFILE d)	3 38	E k) ENGLISH CORRELATIVES	a) HENRY MUDARCZYK
2 32	t) THE	TROUBLE TREE d) COMPUTER	4 30	THER EDIT AIDE d) LETTERS	a) JACK HARDMAN n) MIKE R
4 18	EDITOR ABORT, CORRECTING	TYPOS a) MARK JOHNSON	2 35	PUTING MODULOS d) LETTERS	a) JOSEPH ZELIGS
2 22	FEATURES, LOCATE, EDITOR	U, PQ VS. PGN, PAGE NUMBE	4 18	R ABORT, CORRECTING TYPOS	a) MARK JOHNSON
3 31	GECCOLLECT, BAUD SWITCHES,	US08E	3 6	k) DEFAULT PROC ARGUMENTS	a) MICHAEL ROSSETTI
4 12	t)	UNCOMPILING, PART 1 d) UT	3 36	/PLOT UPGRADES d) LETTERS	a) MIKE SCHELLENBACH
4 25	t) AN	UNDOCUMENTED EDITOR CAPAB	1 43	) KDK ENTERPRISES p) GAAP	c) ACCOUNTING PACKAGE
2 22	B, PQ-COMPILE STON, PAGE,	UNDOCUMENTED FEATURES, LO	2 4	) KDK ENTERPRISES p) GAAP	c) ACCOUNTING PACKAGE
3 36	t) ACCU/PLOT	UPGRADES d) LETTERS a) MI	3 7	D ASSET MANAGEMENT SYSTEM	c) ACCOUNTING PACKAGE
1 44	) AD v) SHEBASTA ASSOC c)	USED HARDWARE	3 41	) KDK ENTERPRISES p) GAAP	c) ACCOUNTING PACKAGE
3 8	d) AD v) EASTCOMP II c)	USED HARDWARE	4 8	) KDK ENTERPRISES p) GAAP	c) ACCOUNTING PACKAGE
3 11	d) AD v) SHEBASTA c)	USED HARDWARE	4 33	D ASSET MANAGEMENT SYSTEM	c) ACCOUNTING PACKAGE
3 37	AD v) ZENTEC p) REFLEX c)	USED HARDWARE	4 41	EMS p) FIXED ASSET MODULE	c) ACCOUNTING PACKAGE
4 11	d) AD v) SHEBASTA c)	USED HARDWARE	1 43	AD v) SEMAPHORE p) PRAGMA	c) ADVERTISING SPACE
4 34	O p) MICRODATA REALITY c)	USED HARDWARE	2 8	AD v) SEMAPHORE p) PRAGMA	c) ADVERTISING SPACE
4 34	AD v) ZENTEC p) REFLEX c)	USED HARDWARE	3 30	AD v) SEMAPHORE p) PRAGMA	c) ADVERTISING SPACE
4 34	d) AD v) EASTCOMP II c)	USED HARDWARE	3 39	v) SOFTWARE GROUP p) AIDS	c) APPLICATIONS GENERATOR
1 43	d) AD v) MICRU INTL c)	USER GROUP	3 47	AD v) CHRONON DATA p) HAL	c) APPLICATIONS GENERATOR
4 25	d) LOCAL	USER GROUPS k) DELAWARE V	4 7	v) SOFTWARE GROUP p) AIDS	c) APPLICATIONS GENERATOR
1 18	t) MOLECTRON d)	USER PROFILE k) MANUFACTU	4 48	AD v) CHRONON DATA p) HAL	c) APPLICATIONS GENERATOR
3 17	RAINBOW NATURAL FOODS d)	USER PROFILE k) MICRODATA	1 37	TRIBUTORS p) IBM SERIES/1	c) COMPUTER SYSTEM
4 14	t) PACIFIC VALLEY BANK d)	USER PROFILE k) MICRODATA	1 42	UTER SYSTEMS p) EVOLUTION	c) COMPUTER SYSTEM
2 16	t) GALLINSKI HAMBURG d)	USER PROFILE k) REALITY T	1 47	D v) MICRODATA p) REALITY	c) COMPUTER SYSTEM
4 24	K, RICHARD CHESSE, RICHARD	USRY, BRIAN EHLERS, W SHA	3 33	d) AD v) PICK SYSTEMS	c) CONTEST
1 9	ERING STATEMENT LABELS d)	UTILITIES	3 5	REALITY TECHNICAL JOURNAL	c) DOCUMENTATION
3 12	REFERENCING Q-POINTERS d)	UTILITIES	4 5	REALITY TECHNICAL JOURNAL	c) DOCUMENTATION
4 12	t) UNCOMPILING, PART 1 d)	UTILITIES k) BASIC OBJECT	4 35	TENS p) PICK POCKET GUIDE	c) DOCUMENTATION
2 11	TRIM.DELIM AND PROFILE d)	UTILITIES k) FILE CONSTRU	3 44	TERPRISES p) ACCU-PLOT II	c) GRAPHICS SOFTWARE
1 15	t)	VANILLA, THE NO-FRILLS MA	4 44	TERPRISES p) ACCU-PLOT II	c) GRAPHICS SOFTWARE
2 27	t)	VANILLA, THE NO-FRILLS MA	1 41	RE p) CONSULTING SERVICES	c) MANUFACTURING
3 30	t)	VANILLA, THE NO-FRILLS MA	2 48	RE p) CONSULTING SERVICES	c) MANUFACTURING
4 32	t)	VANILLA, THE NO-FRILLS MA	3 33	PT p) CONSULTING SERVICES	c) MANUFACTURING
2 10	DIE RANDS, DAN OLDS, BILL	VAUGHN, JIM LEGGETT	3 48	RE p) CONSULTING SERVICES	c) MANUFACTURING
2 25	d) SUBSCRIBERS n) MICHAEL	VERKLER, LESLIE KEENE, BI	4 41	ATA ASSOC p) THE SIDEKICK	c) NETWORK HARDWARE/SOFTW
3 8	ATIN, ROBERT MURRAY, JOHN	VETTER, JOEL NIRENBERG, B	1 46	d) AD v) COSMOS p) R/NET	c) NETWORK SOFTWARE
4 18	S VS. PRIME VS. MICRODATA	VS. ADDS n) ROGER HARPEL,	3 37	d) AD v) COSMOS p) R/NET	c) NETWORK SOFTWARE
2 31	AILABLE DISK SPACE, PRISM	VS. ADDS, WHERE LINKS, IT	4 43	d) AD v) COSMOS p) R/NET	c) NETWORK SOFTWARE
3 28	COMMAND FILES k) ENGLISH	VS. BASIC	4 2	D v) COSMOS p) REVELATION	c) OPERATING SYSTEM
2 20	t) LOOP	VS. LOCATE d) BENCHMARKS	3 33	D v) BURNS & ASSOC p) SAM	c) PERFORMANCE MONITOR
4 18	TICLE k) COSMOS VS. PRIME	VS. MICRODATA VS. ADDS n)	4 41	D v) BURNS & ASSOC p) SAM	c) PERFORMANCE MONITOR
2 22	RES, LOCATE, EDITOR U, PQ	VS. PGN, PAGE NUMBERS, CH	2 8	d) AD v) IRVINE COMPUTER	c) PERIPHERAL HARDWARE
4 18	IONS d) ARTICLE k) COSMOS	VS. PRIME VS. MICRODATA V	3 33	d) AD v) IRVINE COMPUTER	c) PERIPHERAL HARDWARE
4 17	t) LEFT	VS. RIGHT JUSTIFICATION d	3 8	v) COMPUTER DISTRIBUTORS	c) PERSONNEL WANTED
2 31	UMBERS, BREAK-ON, SYSPROG	VS. SP, CREATING FILES	2 48	COMMUNICATIONS CONTROLLER	c) PORT HARDWARE
2 43	ERS n) JOHN HARDMAN, WARD	VUILLEMOT, SERGE KRISIUK,	2 8	) AUTOMATIC LINE OPERATOR	c) PRINTER HARDWARE
3 8	DISTRIBUTORS c) PERSONNEL	WANTED	2 8	TIC PROGRAMMING p) WIZARD	c) PROGRAM GENERATOR
3 33	T p) MICRODATA REALITY c)	WANTED TO BUY	3 8	TIC PROGRAMMING p) WIZARD	c) PROGRAM GENERATOR
4 3	ARSEN, STEVE MATTSON, TOM	WELCH, PETER COY, CHERYL	4 41	TIC PROGRAMMING p) WIZARD	c) PROGRAM GENERATOR
1 3	t)	WELCOME TO PRAGMA d) EDIT	4 1	IVE SYSTEMS p) SCREEN-GEN	c) SCREEN PROCESSOR
2 31	SK SPACE, PRISM VS. ADDS,	WHERE LINKS, ITEM LOCKS,	1 45	R, REPORT-GEN, SCREEN-GEN	c) SOFTWARE TOOLS
4 29	TWARE d) ARTICLE n) ROBIN	WHITE, JIM RENALDS, TERRY	1 24	AD v) SEMAPHORE p) CARGO	c) SPREADSHEET
3 8	ERG, BRUCE CORBETT, FRANK	WILEY	2 5	D v) AUROTECH p) AUROPLAN	c) SPREADSHEET
4 18	d)	WISH LIST k) COPY NULL SU	2 41	VE SYSTEMS p) COMPU-SHEET	c) SPREADSHEET
3 27	d)	WISH LIST k) HEADING EXPR	2 46	AD v) SEMAPHORE p) CARGO	c) SPREADSHEET
2 22	d)	WISH LIST k) PASSWORDS, D	3 2	D v) AUROTECH p) AUROPLAN	c) SPREADSHEET
1 27	d)	WISH LIST k) SLEEP, BREAK	3 47	AD v) SEMAPHORE p) CARGO	c) SPREADSHEET
3 35	t) SOME	WISHES ARE ANSWERED d) LE	3 1	VE SYSTEMS p) COMPU-SHEET	c) SPREADSHEET

4 31	VE SYSTEMS	p)	COMPU-SHEET	c)	SPREADSHEET	1 45	AD v)	INTERACTIVE SYSTEMS	p)	TCL-AUDITOR, REPORT-GE		
4 47	D v)	AUROTECH	p)	AUROPLAN	c)	SPREADSHEET	4 41	d) AD v)	ANSDATA ASSOC	p)	THE SIDEKICK c) NETWORK	
1 3	AD v)	SEMAPHORE	p)	PRAGMA	c)	SUBSCRIPTION	2 8	v)	AUTOMATIC PROGRAMMING	p)	WIZARD c) PROGRAM GENE	
1 45	AD v)	SEMAPHORE	p)	PRAGMA	c)	SUBSCRIPTION	3 8	v)	AUTOMATIC PROGRAMMING	p)	WIZARD c) PROGRAM GENE	
1 46	AD v)	SEMAPHORE	p)	PRAGMA	c)	SUBSCRIPTION	4 41	v)	AUTOMATIC PROGRAMMING	p)	WIZARD c) PROGRAM GENE	
2 3	AD v)	SEMAPHORE	p)	PRAGMA	c)	SUBSCRIPTION	3 44	d)	AD	v)	ACCU/SOFT ENTERPRISES	
2 33	AD v)	SEMAPHORE	p)	PRAGMA	c)	SUBSCRIPTION	4 44	d)	AD	v)	ACCU/SOFT ENTERPRISES	
2 43	AD v)	SEMAPHORE	p)	PRAGMA	c)	SUBSCRIPTION	3 33	d)	AD	v)	ADDEPT p) CONSULTING S	
3 37	AD v)	SEMAPHORE	p)	PRAGMA	c)	SUBSCRIPTION	3 33	d)	AD	v)	ALPHATEST p) MICRODATA	
4 24	AD v)	SEMAPHORE	p)	PRAGMA	c)	SUBSCRIPTION	4 41	d)	AD	v)	ANSDATA ASSOC p) THE S	
1 44	d)	AD v)	SHEBASTA ASSOC	c)	USED HARDWARE	2 5	d)	AD	v)	AUROTECH p) AUROPLAN c		
3 8	d)	AD v)	EASTCOMP II	c)	USED HARDWARE	3 2	d)	AD	v)	AUROTECH p) AUROPLAN c		
3 11	d)	AD v)	SHEBESTA	c)	USED HARDWARE	4 47	d)	AD	v)	AUROTECH p) AUROPLAN c		
3 37	d)	AD v)	ZENTEC p) REFLEX	c)	USED HARDWARE	2 8	d)	AD	v)	AUTOMATIC PROGRAMMING		
4 11	d)	AD v)	SHEBESTA	c)	USED HARDWARE	3 8	d)	AD	v)	AUTOMATIC PROGRAMMING		
4 34	ILLO p)	MICRODATA REALITY	c)	USED HARDWARE	4 41	d)	AD	v)	AUTOMATIC PROGRAMMING			
4 34	d)	AD v)	ZENTEC p) REFLEX	c)	USED HARDWARE	3 33	d)	AD	v)	BURNS & ASSOC p) SAM c		
4 34	d)	AD v)	EASTCOMP II	c)	USED HARDWARE	4 41	d)	AD	v)	BURNS & ASSOC p) SAM c		
1 43	d)	AD v)	MICRU INTL	c)	USER GROUP	3 47	d)	AD	v)	CHRONON DATA p) HAL c)		
3 33	TEST p)	MICRODATA REALITY	c)	WANTED TO BUY	4 48	d)	AD	v)	CHRONON DATA p) HAL c)			
3 44	v)	ACCU/SOFT ENTERPRISES	p)	ACCU-PLOT II c) GRAPHI	3 8	d)	AD	v)	COMPUTER DISTRIBUTORS			
4 44	v)	ACCU/SOFT ENTERPRISES	p)	ACCU-PLOT II c) GRAPHI	1 37	d)	AD	v)	COMPUTER DISTRIBUTORS			
3 39	d)	AD v)	SOFTWARE GROUP	p)	AIDS c) APPLICATIONS G	1 46	d)	AD	v)	COSMOS p) R/NET c) NET		
4 7	d)	AD v)	SOFTWARE GROUP	p)	AIDS c) APPLICATIONS G	3 37	d)	AD	v)	COSMOS p) R/NET c) NET		
2 5	d)	AD v)	AUROTECH	p)	AUROPLAN c) SPREADSHEE	4 43	d)	AD	v)	COSMOS p) R/NET c) NET		
3 2	d)	AD v)	AUROTECH	p)	AUROPLAN c) SPREADSHEE	4 2	d)	AD	v)	COSMOS p) REVELATION c		
4 47	d)	AD v)	AUROTECH	p)	AUROPLAN c) SPREADSHEE	2 8	d)	AD	v)	DELTA DATA CONSULTANTS		
2 8	v)	DELTA DATA CONSULTANTS	p)	AUTOMATIC LINE OPERATO	4 41	d)	AD	v)	DIAGENES COMPUTER SYST			
1 24	d)	AD v)	SEMAPHORE	p)	CARGO c) SPREADSHEET	1 48	d)	AD	v)	DYNATRONICS p) COMMUNI		
2 46	d)	AD v)	SEMAPHORE	p)	CARGO c) SPREADSHEET	3 8	d)	AD	v)	EASTCOMP II c) USED HA		
3 47	d)	AD v)	SEMAPHORE	p)	CARGO c) SPREADSHEET	4 34	d)	AD	v)	EASTCOMP II c) USED HA		
1 48	d)	AD v)	DYNATRONICS	p)	COMMUNICATIONS CONTROL	1 42	d)	AD	v)	EVOLUTION COMPUTER SYS		
2 41	AD v)	INTERACTIVE SYSTEMS	p)	COMPU-SHEET c) SPREADS	3 5	d)	AD	v)	INFORMATION RESEARCH O			
3 1	AD v)	INTERACTIVE SYSTEMS	p)	COMPU-SHEET c) SPREADS	4 5	d)	AD	v)	INFORMATION RESEARCH O			
4 31	AD v)	INTERACTIVE SYSTEMS	p)	COMPU-SHEET c) SPREADS	2 41	d)	AD	v)	INTERACTIVE SYSTEMS p)			
1 41	d)	AD v)	SEMAPHORE	p)	CONSULTING SERVICES c)	3 1	d)	AD	v)	INTERACTIVE SYSTEMS p)		
2 48	d)	AD v)	SEMAPHORE	p)	CONSULTING SERVICES c)	4 31	d)	AD	v)	INTERACTIVE SYSTEMS p)		
3 33	d)	AD v)	ADDEPT	p)	CONSULTING SERVICES c)	4 1	d)	AD	v)	INTERACTIVE SYSTEMS p)		
3 48	d)	AD v)	SEMAPHORE	p)	CONSULTING SERVICES c)	1 45	d)	AD	v)	INTERACTIVE SYSTEMS p)		
1 42	VOLUTION COMPUTER SYSTEMS	p)	EVOLUTION c) COMPUTER			2 8	d)	AD	v)	IRVINE COMPUTER c) PER		
3 7	d)	AD v)	MULTIDATA	p)	FIXED ASSET MANAGEMENT	3 33	d)	AD	v)	IRVINE COMPUTER c) PER		
4 33	d)	AD v)	MULTIDATA	p)	FIXED ASSET MANAGEMENT	1 43	d)	AD	v)	KDK ENTERPRISES p) GAA		
4 41	DIAGENES COMPUTER SYSTEMS	p)	FIXED ASSET MODULE c)			2 4	d)	AD	v)	KDK ENTERPRISES p) GAA		
1 43	d)	AD v)	KDK ENTERPRISES	p)	GAAP c) ACCOUNTING PAC	3 41	d)	AD	v)	KDK ENTERPRISES p) GAA		
2 4	d)	AD v)	KDK ENTERPRISES	p)	GAAP c) ACCOUNTING PAC	4 8	d)	AD	v)	KDK ENTERPRISES p) GAA		
3 41	d)	AD v)	KDK ENTERPRISES	p)	GAAP c) ACCOUNTING PAC	1 47	d)	AD	v)	MICRODATA p) REALITY c		
4 8	d)	AD v)	KDK ENTERPRISES	p)	GAAP c) ACCOUNTING PAC	1 43	d)	AD	v)	MICRU INTL c) USER GRO		
3 47	d)	AD v)	CHRONON DATA	p)	HAL c) APPLICATIONS GE	3 7	d)	AD	v)	MULTIDATA p) FIXED ASS		
4 48	d)	AD v)	CHRONON DATA	p)	HAL c) APPLICATIONS GE	4 33	d)	AD	v)	MULTIDATA p) FIXED ASS		
1 37	v)	COMPUTER DISTRIBUTORS	p)	IBM SERIES/1 c) COMPUT	4 34	d)	AD	v)	PAULA MANNILLO p) MICR			
4 34	d)	AD v)	PAULA MANNILLO	p)	MICRODATA REALITY c) U	3 33	d)	AD	v)	PICK SYSTEMS c) CONTE		
3 33	d)	AD v)	ALPHATEST	p)	MICRODATA REALITY c) W	4 35	d)	AD	v)	PICK SYSTEMS p) PICK P		
4 35	d)	AD v)	PICK SYSTEMS	p)	PICK POCKET GUIDE c) D	1 24	d)	AD	v)	SEMAPHORE p) CARGO c)		
1 43	d)	AD v)	SEMAPHORE	p)	PRAGMA c) ADVERTISING	2 46	d)	AD	v)	SEMAPHORE p) CARGO c)		
2 8	d)	AD v)	SEMAPHORE	p)	PRAGMA c) ADVERTISING	3 47	d)	AD	v)	SEMAPHORE p) CARGO c)		
3 30	d)	AD v)	SEMAPHORE	p)	PRAGMA c) ADVERTISING	1 41	d)	AD	v)	SEMAPHORE p) CONSULTIN		
1 3	d)	AD v)	SEMAPHORE	p)	PRAGMA c) SUBSCRIPTION	2 48	d)	AD	v)	SEMAPHORE p) CONSULTIN		
1 45	d)	AD v)	SEMAPHORE	p)	PRAGMA c) SUBSCRIPTION	3 48	d)	AD	v)	SEMAPHORE p) CONSULTIN		
1 46	d)	AD v)	SEMAPHORE	p)	PRAGMA c) SUBSCRIPTION	1 43	d)	AD	v)	SEMAPHORE p) PRAGMA c)		
2 3	d)	AD v)	SEMAPHORE	p)	PRAGMA c) SUBSCRIPTION	2 8	d)	AD	v)	SEMAPHORE p) PRAGMA c)		
2 33	d)	AD v)	SEMAPHORE	p)	PRAGMA c) SUBSCRIPTION	3 30	d)	AD	v)	SEMAPHORE p) PRAGMA c)		
2 43	d)	AD v)	SEMAPHORE	p)	PRAGMA c) SUBSCRIPTION	1 3	d)	AD	v)	SEMAPHORE p) PRAGMA c)		
3 37	d)	AD v)	SEMAPHORE	p)	PRAGMA c) SUBSCRIPTION	1 45	d)	AD	v)	SEMAPHORE p) PRAGMA c)		
4 24	d)	AD v)	SEMAPHORE	p)	PRAGMA c) SUBSCRIPTION	1 46	d)	AD	v)	SEMAPHORE p) PRAGMA c)		
1 46	d)	AD v)	COSMOS	p)	R/NET c) NETWORK SOFTW	2 3	d)	AD	v)	SEMAPHORE p) PRAGMA c)		
3 37	d)	AD v)	COSMOS	p)	R/NET c) NETWORK SOFTW	2 33	d)	AD	v)	SEMAPHORE p) PRAGMA c)		
4 43	d)	AD v)	COSMOS	p)	R/NET c) NETWORK SOFTW	2 43	d)	AD	v)	SEMAPHORE p) PRAGMA c)		
3 5	TION RESEARCH OF NEW YORK	p)	REALITY TECHNICAL JOUR			3 37	d)	AD	v)	SEMAPHORE p) PRAGMA c)		
4 5	TION RESEARCH OF NEW YORK	p)	REALITY TECHNICAL JOUR			4 24	d)	AD	v)	SEMAPHORE p) PRAGMA c)		
1 47	d)	AD v)	MICRODATA	p)	REALITY c) COMPUTER SY	1 44	d)	AD	v)	SHEBASTA ASSOC c) USED		
3 37	d)	AD v)	ZENTEC	p)	REFLEX c) USED HARDWAR	3 11	d)	AD	v)	SHEBESTA c) USED HARDW		
4 34	d)	AD v)	ZENTEC	p)	REFLEX c) USED HARDWAR	4 11	d)	AD	v)	SHEBESTA c) USED HARDW		
4 2	d)	AD v)	COSMOS	p)	REVELATION c) OPERATIN	3 39	d)	AD	v)	SOFTWARE GROUP p) AIDS		
3 33	d)	AD v)	BURNS & ASSOC	p)	SAM c) PERFORMANCE MON	4 7	d)	AD	v)	SOFTWARE GROUP p) AIDS		
4 41	d)	AD v)	BURNS & ASSOC	p)	SAM c) PERFORMANCE MON	3 37	d)	AD	v)	ZENTEC p) REFLEX c) US		
4 1	AD v)	INTERACTIVE SYSTEMS	p)	SCREEN-GEN c) SCREEN P	4 34	d)	AD	v)	ZENTEC p) REFLEX c) US			

games

Amazing!

You've been wandering through the maze for over three hours now, trying to find the exit. Your CRT is displaying the view from your current position inside the maze:



You are looking down a corridor that ends up turning right, but there is also a passage off to the left and an even nearer hall off to the right. Which direction should you try?

For many people, computers are synonymous with games, now that the video game industry has become such a giant. Programmers frequently cut their teeth on small game programs, since such programs are often straight-forward and self-contained without a lot of complicated interfaces to files or other software. And, more than anything, games are simply entertaining and fun.

The Games Department will be making periodic appearances in *Pragma*. If you have a game program you would like to share, send it in for publication. If there's anything the *Pragma* staff has time to do, it's "performing rigorous software quality assurance" (in other words, trying out a new game on the computer).

AMAZING is a program that displays views like the one on the left to try and help you find your way through mazes that you or anyone else can design.

Before you can explore a maze with AMAZING, the walls and passages of the maze must be input using the MAKE.MAZE program, which is listed on page 53 and followed by a sample of an actual maze. The equate on line 2 of MAKE.MAZE limits the number of columns and rows in the maze, but you may adjust cmax and rmax to any number within the limits of the display screen if you want to design larger mazes.

MAKE.MAZE begins by displaying every possible wall in a maze. Use numeric keys to drive the cursor around while using the space bar to blank out walls and create passages. A non-blank key will resurrect a wall. The "F" key will file away the maze and the "X" key will exit the program. A file called MAZES must exist for saving the maze data.

Once a maze has been created with MAKE.MAZE, it can be explored using AMAZING, which is listed beginning on page 54. AMAZING starts you off in the northwest corner of the maze and shows you the view looking east. You can type "N", "S", "E" or "W" to see the view in those directions, or you can use "M" to move in the direction you're already looking. Entering "Q" will let you quit.

AMAZING works by assuming that there is always at least one wall blocking the view within four units of the observer. It is up to the maze designer to insure there are no corridors longer than four units or huge, cavernous rooms in the maze. The EDGES item on page 54, which must be in the dictionary of the MAZES file, is a list of the endpoint co-ordinates of the 82 different possible line segments (wall edges) that may be displayed. The OFFSETS item, also for the MAZES dictionary, tells AMAZING all possible wall positions within four units when looking north.

Subroutine 20 at line 65 draws a line on the screen using column and row positions from EDGES, and is the primary candidate for optimization if you want to speed up AMAZING. But if the object is to make AMAZING more fun, how about changing the programs to allow a pile of gold to be hidden by the maze designer so that it can be searched for and displayed by the maze explorer?

P

**OFFICE
OFFICE
OFFICE
OFFICE**

**MULE
MULE
MULE
MULE**

OFFICE MULE DATABASE MANAGEMENT LANGUAGE

- ☐ Up to Twice the Speed of BASIC for Data Entry
- ☐ Handles Data Entry, Purge, Forms and Reports
- ☐ Easy to Use
- ☐ Completely Self-Documenting
- Multiple Screen Capabilities ☐
- Many Unique Features ☐
- Help Screen Automatically Generated for Every Prompt ☐
- Compile Speed Comparable to BASIC ☐

OFFICE MULE DATABASE BUILDER

- ☐ Completely Menu Driven
- ☐ Builds Pick Dictionaries for English
- Edits Mistakes from File ☐

OFFICE MULE FINANCIAL MODELING

- ☐ Handles Any Size Spreadsheet
- ☐ Easy Spreadsheet Handling
- Fully Integrated to Database Builder ☐

OFFICE MULE EXECUTIVE COORDINATOR

- ☐ Records Phone Messages
- ☐ Schedules Appointments and Meetings
- ☐ Maintains "Things to Do" List
- Use as an Address Book ☐
- Files Memos ☐
- Electronic Mail ☐

SYSTEM DEVELOPMENT CONSULTING PACKAGES

LAZAROV SOFTWARE SYSTEMS, INC.
12201 E. ARIZONA DR.
AURORA, CO 80012
303-755-1451
TELEX: 45-885 TAS/DVR




```

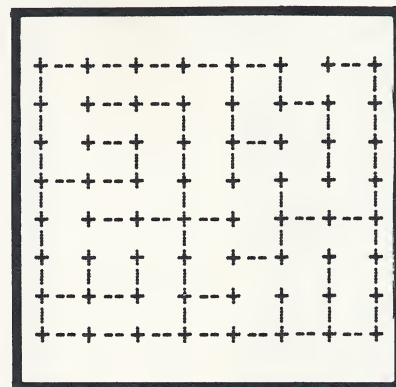
MAKE.MAZE
001 EQU clear TO CHAR(12)
002 EQU cmax TO 15, rmax TO 15 ;# Must be odd
003 OPEN "MAZES" ELSE STOP "No MAZES!"
004 PRINT clear:
005 PRINT "NAME OF MAZE": ; INPUT NAME
006 NEW = 0
007 READ MAZE FROM NAME ELSE NEW = 1
008 IF NEW THEN
009 MAZE = ""
010 PRINT "COLUMNS (1-":cmax:"):":
011 INPUT COLS
012 PRINT "ROWS (1-":rmax:"):":
013 INPUT ROWS
014 END ELSE
015 COLS = COUNT(MAZE,CHAR(254))+1
016 ROWS = COUNT(MAZE(1),CHAR(253))+1
017 END
018 * Must be odd and within bounds
019 COLS = COLS + 1 - MOD(COLS,2)
020 ROWS = ROWS + 1 - MOD(ROWS,2)
021 IF COLS < 3 THEN COLS = 3
022 IF ROWS < 3 THEN ROWS = 3
023 IF COLS > cmax THEN COLS = cmax
024 IF ROWS > rmax THEN ROWS = rmax
025 PRINT clear:
026 FOR C = 1 TO COLS
027   FOR R = 1 TO ROWS
028     * Display default or existing maze
029     IF NEW THEN GOSUB 300 ELSE GOSUB 250
030   NEXT R
031 NEXT C
032 C = 1 ; R = 1 ; PROMPT ""
033 LOOP
034 PRINT @(C,R):
035 INPUT SYMBOL,1:
036 UNTIL (SYMBOL="F") OR (SYMBOL="X") DO
037   GOSUB 250 ;# Restore screen char
038 BEGIN CASE
039 CASE SYMBOL = 2 ; R=R+1 ; GOSUB 100
040 CASE SYMBOL = 4 ; C=C-1 ; GOSUB 100
041 CASE SYMBOL = 6 ; C=C+1 ; GOSUB 100
042 CASE SYMBOL = 8 ; R=R-1 ; GOSUB 100
043 CASE SYMBOL = " "
044   GOSUB 400 ;# Check if at +
045   IF NOT(COL.000 & ROW.000) THEN GOSUB 200
046 CASE 1 ; GOSUB 300 ;# Make wall
047 END CASE
048 REPEAT
049 IF SYMBOL = "F" THEN WRITE MAZE ON NAME
050 PRINT clear:
051 STOP
052 *

```

```

053 100 * Stay within bounds
054 IF C < 1 THEN C = 1
055 IF R < 1 THEN R = 1
056 IF C > COLS THEN C = COLS
057 IF R > ROWS THEN R = ROWS
058 RETURN
059 *
060 200 * Save maze symbol
061 MAZE(C,R) = SYMBOL
062 250 * Display maze char
063 PRINT @(C,R):MAZE(C,R):
064 RETURN
065 *
066 300 * Build default walls
067 GOSUB 400 ;# Check if at +
068 BEGIN CASE
069 CASE COL.000 AND ROW.000
070   SYMBOL = "+":
071 CASE COL.000
072   SYMBOL = "1":
073 CASE ROW.000
074   SYMBOL = "-":
075 CASE 1
076   SYMBOL = " "
077 END CASE
078 GOSUB 200 ;# Save maze char
079 RETURN
080 *
081 400 * Check if odd col or row
082 COL.000 = MOD(C,2)
083 ROW.000 = MOD(R,2)
084 RETURN
085 *
086 END

```



A TYPICAL MAZE

MAKE.MAZE LISTING

EDGES

001 0111713	042 721917919
002 721317911	043 3119131113
003 0131713	044 4719147113
004 71317213	045 011317113
005 721317913	046 7113115113
006 71317119	047 15113123113
007 71311515	048 23113131113
008 631517213	049 31113147113
009 7213172119	050 47113155113
010 0151715	051 55113163113
011 71511515	052 63113172113
012 151516315	053 72113179113
013 631517215	054 011517114
014 721517915	055 7114115113
015 1515115117	056 23115131113
016 151512317	057 47113155115
017 551716315	058 63113172114
018 6315163117	059 72114179115
019 0171717	060 011517115
020 71711517	061 7115115115
021 151712317	062 15115123115
022 231715517	063 23115155115
023 551716317	064 55115163115
024 631717217	065 63115172115
025 721717917	066 72115179115
026 0171718	067 15117123115
027 71811519	068 55115163117
028 2317123115	069 011717117
029 231713119	070 7117115117
030 471915517	071 15117163117
031 5517155115	072 63117172117
032 631917218	073 72117179117
033 721817917	074 7119115117
034 0191719	075 63117172119
035 71911519	076 011917119
036 151912319	077 7119172119
037 231913119	078 72119179119
038 311914719	079 012117119
039 471915519	080 72119179121
040 551916319	081 1519115113
041 631917219	082 6319163113

EDGES ITEM

OFFSETS

001 -41-7	013 -11-4
002 -21-7	014 11-4
003 01-7	015 -21-3
004 21-7	016 01-3
005 41-7	017 21-3
006 -31-6	018 -11-2
007 -11-6	019 11-2
008 11-6	020 -21-1
009 31-6	021 01-1
010 -21-5	022 21-1
011 01-5	023 -110
012 21-5	024 110

OFFSETS ITEM

AMAZING

```

001 DIM W(24) :# Total possible walls in view
002 EGU clear TO CHAR(12)
003 OPEN "MAZES" ELSE STOP "No MAZES!"
004 PRINT clear:
005 LOOP
006 PRINT "NAME OF MAZE": ; INPUT NAME
007 PRINT clear:
008 MISSING = 0
009 READ MAZE FROM NAME ELSE MISSING = 1
010 WHILE MISSING DO
011 PRINT NAME:" not found!"
012 REPEAT
013 COLS = COUNT(MAZE,CHAR(254))+1
014 ROWS = COUNT(MAZE<1>,CHAR(253))+1
015 OPEN "DICT", "MAZES" ELSE STOP "No MAZES!"
016 READ EDGES FROM "EDGES" ELSE STOP "No EDGES!"
017 READ OFFSETS FROM "OFFSETS" ELSE STOP "No OFFSETS!"
018 C = 2 ; R = 2 ; LOOKING = "east"
019 LOOP
020 GOSUB 10 :# Set W(1...24)
021 GOSUB 30 :# Paint screen
022 PRINT @ (0,23):"Looking ":LOOKING:
023 PRINT ". 'M'ove ahead or look ":
024 PRINT "'N', 'S', 'E', 'W' or 'Q'uit":
025 INPUT DIR,1:
026 PRINT clear:
027 UNTIL DIR = "Q" DO
028 BEGIN CASE
029 CASE DIR = "N" ; LOOKING = "north"
030 CASE DIR = "S" ; LOOKING = "south"
031 CASE DIR = "E" ; LOOKING = "east"
032 CASE DIR = "W" ; LOOKING = "west"
033 CASE DIR = "M"
034 IF NOT(W(21)) THEN
035 BEGIN CASE
036 CASE LOOKING = "north"
037 R = R-2 ; IF R < 2 THEN R = 2
038 CASE LOOKING = "south"
039 R = R+2 ; IF R >= ROWS THEN R = ROWS-1
040 CASE LOOKING = "east"
041 C = C+2 ; IF C >= COLS THEN C = COLS-1
042 CASE LOOKING = "west"
043 C = C-2 ; IF C < 2 THEN C = 2
044 END CASE
045 END
046 END CASE
047 REPEAT
048 STOP
049 *
050 10 * Set W(1...24)
051 FOR WALL = 1 TO 24
052 WX = OFFSETS<WALL,1> ; WY = OFFSETS<WALL,2>
053 BEGIN CASE
054 CASE LOOKING = "north" ; LX = C+WX ; LY = R+WY
055 CASE LOOKING = "south" ; LX = C-WX ; LY = R-WY
056 CASE LOOKING = "east" ; LX = C-WY ; LY = R+WX
057 CASE LOOKING = "west" ; LX = C+WY ; LY = R-WX
058 END CASE
059 IF (1<=LX) & (LX<=COLS) & (1<=LY) & (LY<=COLS) THEN
060 W(WALL) = (MAZE<LX,LY> # " ")
061 END ELSE W(WALL) = 0
062 NEXT WALL
063 RETURN
064 *

```

AMAZING LISTING

AMAZING LISTING CONTINUED

```

065 20 * Plot line
066 XFROM = EDGES(L,1) ; YFROM = EDGES(L,2)
067 XTO = EDGES(L,3) ; YTO = EDGES(L,4)
068 XDELTA = XTO-XFROM
069 IF XDELTA # 0 THEN
070   YDELTA = (YTO-YFROM)/XDELTA
071   XDELTA = 1
072   END ELSE YDELTA = 1
073 LOOP
074   YINT = ICONV(YFROM,"MDO")
075   PRINT @(XFROM,YINT):"+":
076   UNTIL (XFROM = XTO) & (YINT = YTO) DO
077     YFROM = YFROM+YDELTA
078     XFROM = XFROM+XDELTA
079 REPEAT
080 RETURN
081 *
082 30 * Paint screen
083 IF W(23) THEN L = 1 ; GOSUB 20 ; L = 79 ; GOSUB 20
084 IF W(24) THEN L = 2 ; GOSUB 20 ; L = 80 ; GOSUB 20
085 IF W(20) & NOT(W(23)) THEN L = 3 ; GOSUB 20 ; L = 76 ; GOSUB 20
086 IF W(22) & NOT(W(24)) THEN L = 5 ; GOSUB 20 ; L = 78 ; GOSUB 20
087 IF W(21) THEN L = 4 ; GOSUB 20 ; L = 77 ; GOSUB 20
088 IF W(18) ! W(21) ! W(23) ! W(20) THEN L = 6 ; GOSUB 20
089 IF W(19) ! W(22) ! W(24) ! W(21) THEN L = 9 ; GOSUB 20
090 IF W(18) & NOT(W(21)) THEN L = 7 ; GOSUB 20 ; L = 74 ; GOSUB 20
091 IF W(19) & NOT(W(21)) THEN L = 8 ; GOSUB 20 ; L = 75 ; GOSUB 20
092 IF W(15) & NOT(W(23)) ! W(20) THEN L = 10 ; GOSUB 20 ; L = 69 ; GOSUB 20
093 IF W(15) & NOT(W(21)) ! W(18) THEN L = 11 ; GOSUB 20 ; L = 70 ; GOSUB 20
094 IF W(16) & NOT(W(21)) THEN L = 12 ; GOSUB 20 ; L = 71 ; GOSUB 20
095 IF W(17) & NOT(W(19)) ! W(21) THEN L = 13 ; GOSUB 20 ; L = 72 ; GOSUB 20
096 IF W(17) & NOT(W(24)) ! W(22) THEN L = 14 ; GOSUB 20 ; L = 73 ; GOSUB 20
097 IF W(10) & NOT(W(23)) ! W(20) ! W(15) THEN L = 19 ; GOSUB 20 ; L = 60 ; GOSUB 20
098 IF W(12) & NOT(W(24)) ! W(22) ! W(17) THEN L = 25 ; GOSUB 20 ; L = 66 ; GOSUB 20
099 IF W(15) ! W(13) ! W(16) ! W(18) & NOT(W(21)) THEN L = 15 ; GOSUB 20
100 IF W(14) ! W(17) ! W(19) ! W(16) & NOT(W(21)) THEN L = 18 ; GOSUB 20
101 IF W(13) & NOT(W(21)) ! W(16) THEN L = 16 ; GOSUB 20 ; L = 67 ; GOSUB 20
102 IF W(14) & NOT(W(21)) ! W(16) THEN L = 17 ; GOSUB 20 ; L = 68 ; GOSUB 20
103 IF W(10) & NOT(W(18)) ! W(15) THEN L = 20 ; GOSUB 20 ; L = 61 ; GOSUB 20
104 IF W(10) & NOT(W(13)) ! W(16) ! W(21) THEN L = 21 ; GOSUB 20 ; L = 62 ; GOSUB 20
105 IF W(12) & NOT(W(14)) ! W(16) ! W(21) THEN L = 23 ; GOSUB 20 ; L = 64 ; GOSUB 20
106 IF W(11) & NOT(W(16)) ! W(21) THEN L = 22 ; GOSUB 20 ; L = 63 ; GOSUB 20
107 IF W(12) & NOT(W(17)) ! W(19) THEN L = 24 ; GOSUB 20 ; L = 65 ; GOSUB 20
108 IF W(6) & NOT(W(23)) ! W(20) ! W(15) ! W(10) THEN L = 26 ; GOSUB 20 ; L = 54 ; GOSUB 20
109 IF W(9) & NOT(W(24)) ! W(22) ! W(17) ! W(12) THEN L = 33 ; GOSUB 20 ; L = 59 ; GOSUB 20
110 IF W(6) & NOT(W(21)) ! W(18) ! W(15) ! W(10) THEN L = 27 ; GOSUB 20 ; L = 55 ; GOSUB 20
111 IF W(9) & NOT(W(21)) ! W(19) ! W(17) ! W(12) THEN L = 32 ; GOSUB 20 ; L = 58 ; GOSUB 20
112 IF W(7) ! W(11) ! W(13) ! W(10) & NOT(W(21)) ! W(16) THEN L = 28 ; GOSUB 20
113 IF W(8) ! W(12) ! W(14) ! W(11) & NOT(W(21)) ! W(16) THEN L = 31 ; GOSUB 20
114 IF W(7) & NOT(W(11)) ! W(16) ! W(21) THEN L = 29 ; GOSUB 20 ; L = 56 ; GOSUB 20
115 IF W(8) & NOT(W(11)) ! W(16) ! W(21) THEN L = 30 ; GOSUB 20 ; L = 57 ; GOSUB 20
116 IF W(1) & NOT(W(6)) ! W(10) ! W(15) ! W(20) ! W(23) THEN L = 34 ; GOSUB 20 ; L = 45 ; GOSUB 20
117 IF W(5) & NOT(W(9)) ! W(12) ! W(17) ! W(22) ! W(24) THEN L = 42 ; GOSUB 20 ; L = 53 ; GOSUB 20
118 IF W(1) & NOT(W(6)) ! W(10) ! W(15) ! W(18) THEN L = 35 ; GOSUB 20 ; L = 46 ; GOSUB 20
119 IF W(5) & NOT(W(9)) ! W(12) ! W(17) ! W(19) THEN L = 41 ; GOSUB 20 ; L = 52 ; GOSUB 20
120 IF W(2) & NOT(W(10)) ! W(13) ! W(16) ! W(21) THEN L = 36 ; GOSUB 20 ; L = 47 ; GOSUB 20
121 IF W(4) & NOT(W(12)) ! W(14) ! W(16) ! W(21) THEN L = 40 ; GOSUB 20 ; L = 51 ; GOSUB 20
122 IF W(2) & NOT(W(7)) ! W(11) ! W(16) ! W(21) THEN L = 37 ; GOSUB 20 ; L = 48 ; GOSUB 20
123 IF W(4) & NOT(W(8)) ! W(11) ! W(16) ! W(21) THEN L = 39 ; GOSUB 20 ; L = 50 ; GOSUB 20
124 IF W(3) & NOT(W(11)) ! W(16) ! W(21) THEN L = 38 ; GOSUB 20 ; L = 49 ; GOSUB 20
125 IF W(2) ! W(3) ! W(7) & NOT(W(11)) ! W(16) ! W(21) THEN L = 43 ; GOSUB 20
126 IF W(3) ! W(4) ! W(8) & NOT(W(11)) ! W(16) ! W(21) THEN L = 44 ; GOSUB 20
127 IF W(1) ! W(2) ! W(6) & NOT(W(10)) ! W(21) THEN L = 81 ; GOSUB 20
128 IF W(4) ! W(5) ! W(9) & NOT(W(12)) ! W(21) THEN L = 82 ; GOSUB 20
129 RETURN
130 *
131 END

```


PRAGMA

207 GRANADA DRIVE
APTOS, CA 95003

BULK RATE
U.S. POSTAGE
PAID
APTOS, CA 95003
Permit No. 67

Address Correction Requested

SEMAPHORE CORPORATION

Provides Software and Support for Microdata Systems

- COMPLETE SPECIFICATION
- ELEGANT DESIGN
- SYSTEMATIC IMPLEMENTATION
- THOROUGH TESTING
- PATIENT USER TRAINING
- ONLINE DOCUMENTATION
- PROVEN POLICIES AND PROCEDURES

SEMAPHORE

SEMAPHORE CORPORATION
207 GRANADA DRIVE
APTOS, CA 95003
408-688-9200

- SPECIALIZING IN MANUFACTURING SYSTEMS:** ENGINEERING • PURCHASING
• RECEIVING • INSPECTION • ORDER ENTRY • SHIPPING • CUSTOMER SERVICE
• PRODUCTION CONTROL • SHOP FLOOR • PAYROLL • PERSONNEL
• RECEIVABLES • PAYABLES • COST ACCOUNTING • GENERAL LEDGER
• FIXED ASSETS • MODELING